

# Manage Memory With VB.NET

www.vbpj.com • www.vcdj.com

Win a **BMW Lease**  
or **Hawaiian**  
**Vacation!**



Web Services Contest  
Details on Page 36

# Visual Studio<sup>TM</sup> Magazine

A FAWCETTE TECHNICAL PUBLICATION

OCTOBER 2001, VOL. 11, NO. 12

## Clear .NET Hurdles

### Scale Up With ADO Connections

### Auto-Document Your VB Projects

### Easy E-Mail With ASP .NET

INCORPORATING

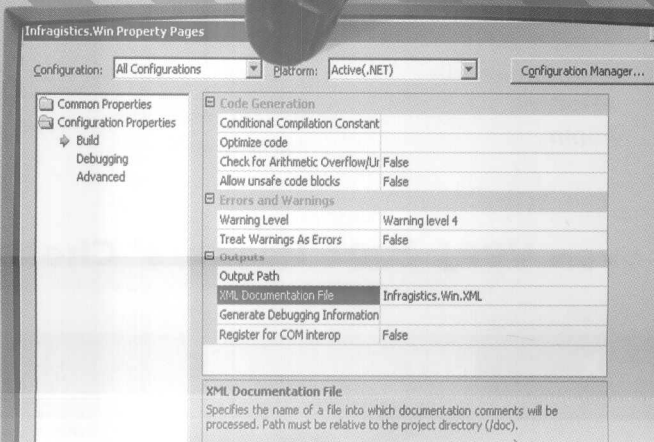
**VISUAL BASIC**  
PROGRAMMER'S JOURNAL

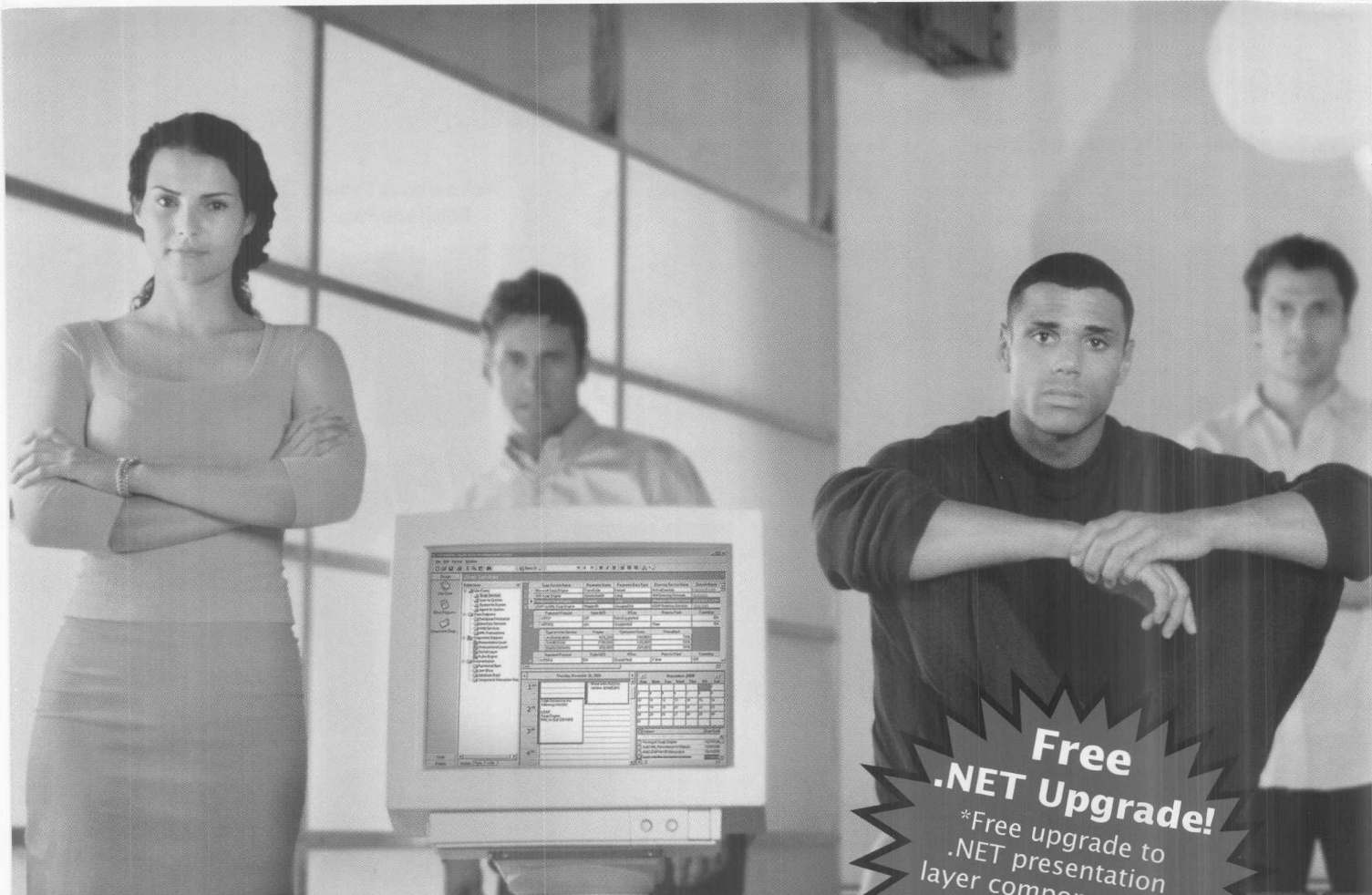
**Visual C++**  
DEVELOPERS JOURNAL

**SQL** **PRO**



U.S. \$6.95 (CANADA \$8.95)





# Suite Rewards.

**Free  
.NET Upgrade!**

*\*Free upgrade to  
.NET presentation  
layer components for  
Subscription and  
Enterprise customers!*

**Infragistics - formed by the merger of Sheridan and ProtoView**

## 45 Award Winning Components included in UltraSuite™ 2.0:

### Grid

- UltraGrid™ 2.0

### Office/Outlook-style UI

- Data Explorer™
- Outlook Bar

### Toolbars

- UltraToolBars™

### Tree

- ActiveTreeView™

### Calendaring & Dates

- Calendar
- DayView
- TaskPad
- DateEdit
- TimeEdit

### Advanced DB Components

- ComboBoxEx
- DropDownEdit
- HotLink
- Spin

- ComboBox
- Listbox

### Tabs

- ActiveTabs™

### Data Input

- Currency
- Numeric
- MaskEdit

### Resizers

- Resizer
- Resizer (works with non-client area)

### Advanced UI Components

- Scroll
- Splash
- Transition
- Marquee
- Picture
- ProgressBar
- Frame
- Option
- Command
- Ribbon
- Check
- Splitter

- Panel
- Dial
- Font Selector
- Line3D
- Multibutton
- Shape3D
- Text3D

### Utility Components

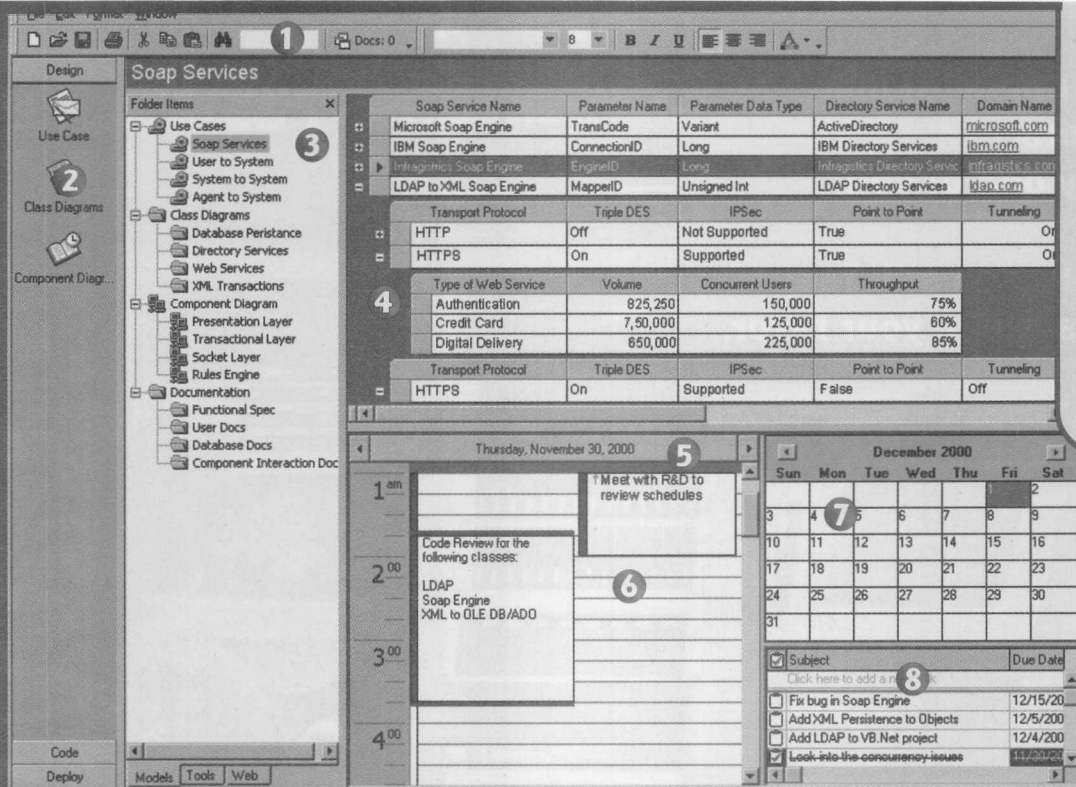
- ScreenPrinter
- PropertyBrowser
- ImageCombo
- ColorCombo

**Recipient of seven VBPI 2001 Readers' Choice Awards**



**Java • ActiveX • .NET**





- 3 TreeView object of the Data Explorer
- 4 UltraGrid control in the right pane of the Data Explorer
- 5 ScheduleX controls in the bottom pane of the Data Explorer
- 6 DayView control (part of ScheduleX)
- 7 Calendar control (part of ScheduleX)
- 8 TaskPad control (part of ScheduleX)

# UltraSuite™ 2.0

We've combined our best-of-breed products into one positively jam-packed solution. Along with the familiar, easy-to-use interfaces of Microsoft Outlook, Microsoft Office and Windows Explorer, UltraSuite provides the most innovative grid available—UltraGrid2.0, and our UltraToolBars is loaded with functionality. With 45 controls, we've included everything necessary to create solutions that look great and run efficiently, faster than ever before. There's nothing like the productivity, flexibility and support of UltraSuite.

You can really sweeten the deal—UltraSuite 2.0 is available with optional Subscription Service, or upgrade to the UltraSuite Enterprise Edition (includes subscription and guaranteed rapid response). Automatically receive all major upgrades (full version releases) and minor updates (enhancements and service releases) during your subscription period.\* Make the most of your investment—you'll find the rewards can be "Suite".

For detailed information, visit  
**[www.infragistics.com](http://www.infragistics.com)**  
 Download Free Trial Version!  
 Order Online! or call 800-231-8588

**Infragistics™**  
 Component powered infrastructure

Copyright 2001 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, UltraSuite, Data Explorer, ActiveTreeView, UltraGrid & UltraToolBars are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

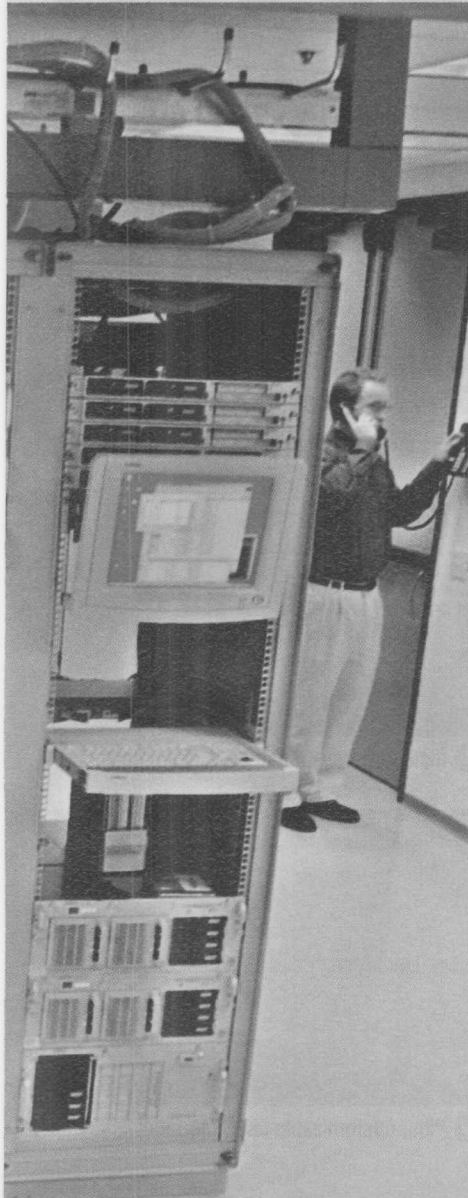
**800-231-8588 • [infragistics.com](http://infragistics.com)**

**14 Web applications.  
79 servers.  
And only one of you.**

**Strangely, you still feel  
the odds are in your favor.**







## MICROSOFT® APPLICATION CENTER 2000

Web server farms have always been known for providing high application availability in a cost-efficient manner. Unfortunately, in the past they have also been known for manageability headaches. But no more: Microsoft Application Center 2000 now makes managing Web applications and groups of Web servers as simple as managing a single server.

Part of the flexible Microsoft .NET Enterprise Server family, Application Center 2000 is built to be the heart of a "scale-out" infrastructure model. "Scaling out" is a flexible approach to scalability that involves deploying Web applications across multiple servers to distribute and handle the workload.

Application Center 2000 makes scaling out easier, with unified Web application and server-farm management that simplifies tasks

### MARKET FACT

"The scalability benefits offered by Application Center made it possible to migrate to a distributed Windows® platform. Application Center gives us on-demand scalability and is a powerful tool for managing Windows 2000-based Web server farms."

—Mike Bodnar, Production Services Manager, Terra Lycos

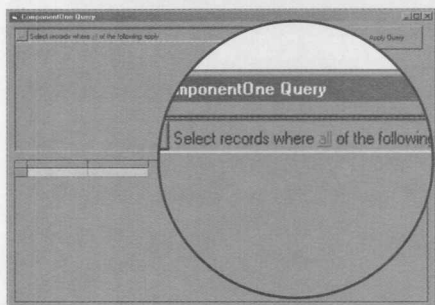
like cluster management and application deployment. Plus, it makes it easy to achieve capacity on demand through automatic replication of applications when you add servers or make changes to existing applications. But simplicity is not all you get:

Application Center 2000 offers increased uptime through dynamic load balancing and by having no single point of failure.

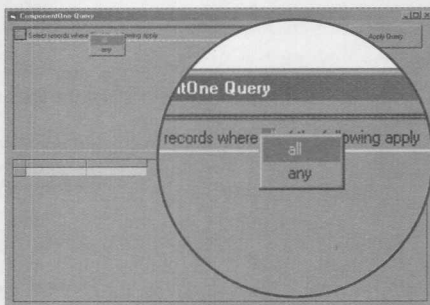
In sum, Application Center 2000 removes the hassle of managing multiple servers separately, while providing the availability and cost-efficiency of a "scale-out" model. Find out how to keep the odds in your favor: visit [microsoft.com/applicationcenter](http://microsoft.com/applicationcenter) **Software for the Agile Business.**

**Microsoft®**

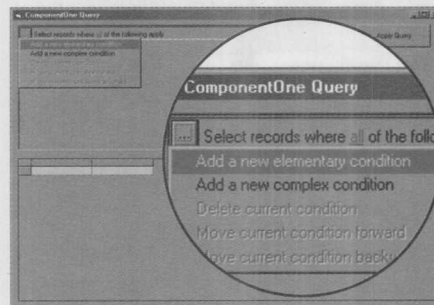




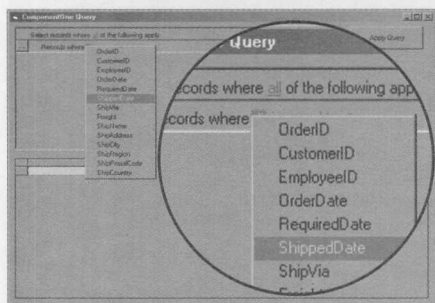
1 Ad hoc end-user queries...



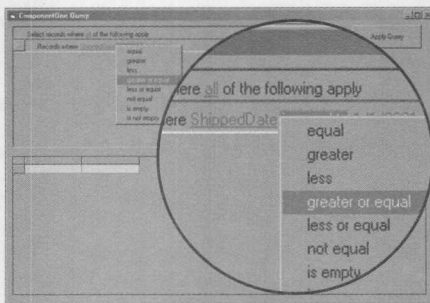
2 ...are a "point-and-click" away!



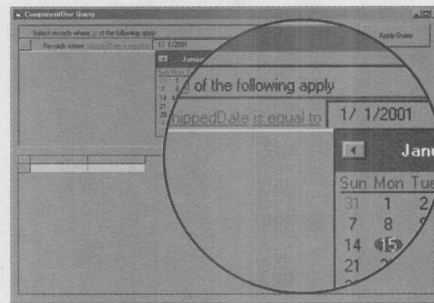
3 End-users simply choose conditions...



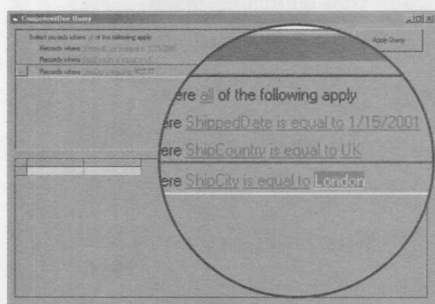
4 ...via drop-downs...



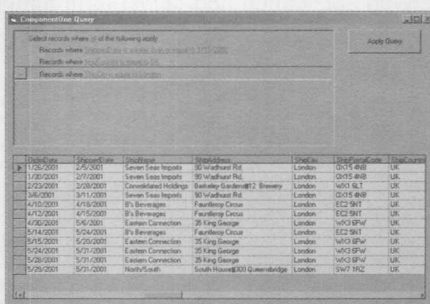
5 ...to quickly and easily set criteria.



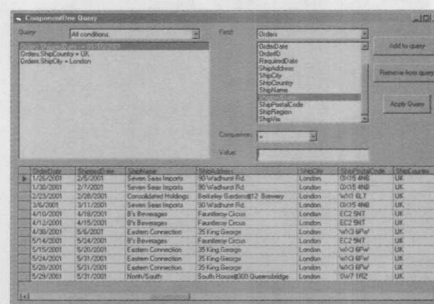
6 Includes easy-to-use drop-down calendar!



7 The completed ad hoc end-user query...



8 ...and the results!



9 Plus, customizable user interface!

## Ask and you shall receive. Ad hoc querying at your service.



Now you can quickly and easily include ad hoc query functionality in your applications and make it directly available to end-users! ComponentOne Query is an ActiveX® control that collects end-user input and generates an SQL query or a Filter condition at run time—no more preparing all imaginable queries in advance at design time! Features include: Powerful query engine; Database structure import; Customizable user interface (via templates or programmatically); Custom data editors; Query generation control; Lookup lists; Multi-table views; Calculated fields; Join relationships and join groups; and more!

Trust the performance, support, and value you can only get from ComponentOne tools. Download your eval copy today!

TAKE ADVANTAGE OF OUR FABULOUS LIMITED TIME OFFERS:

Full Version.....\$399.<sup>95</sup> Premium Support with Subscription.....\$299.<sup>95</sup>

# ComponentOne

www.componentone.com

East Coast Office: 1.800.858.2739 or 1.412.681.4343 • West Coast Office: 1.888.228.4839 or 1.510.496.3240

ComponentOne™ LLC • 4516 Henry Street • Pittsburgh, PA 15213 • USA

© 2001 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.

Now Incorporating

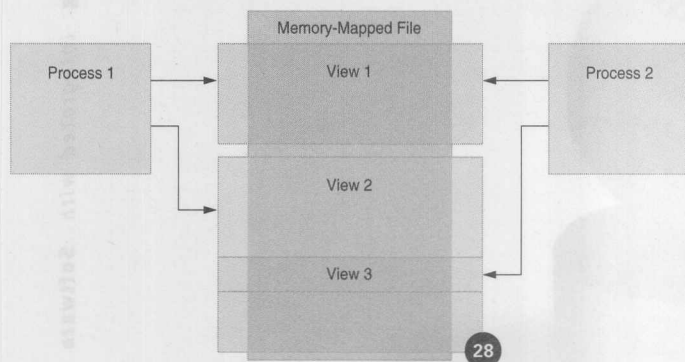
Visual Basic Programmer's Journal • Visual C++ Developers Journal • SQL Pro

## FEATURES

## 28 Pointers Add Power and Safety

Bid farewell to CopyMemory and learn how VB.NET makes direct memory manipulation a breeze.

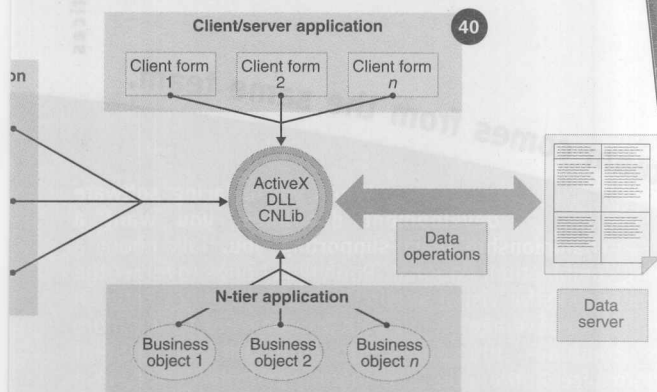
by Robert Teixeira



## 40 Enhance Performance With ADO Connection Objects

Encapsulating an ADO connection inside a COM object lets your application perform better and scale more easily. Here's how to use ADO Connection objects in COM and MTS.

by Dimitrios Tsonis

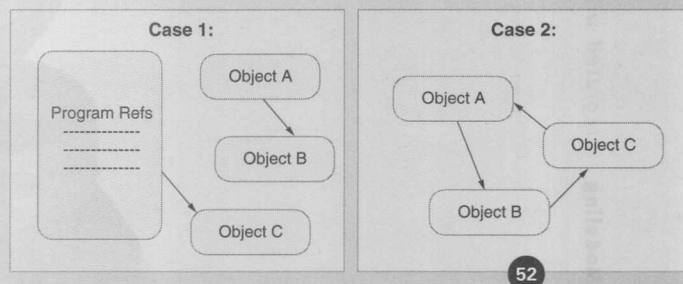


## Visual C++ Developers Journal

## 52 Clear Common C# Hurdles

Learn how to avoid some of the most common gotchas when working with C# from two developers who have been using the language for real development for more than a year and a half.

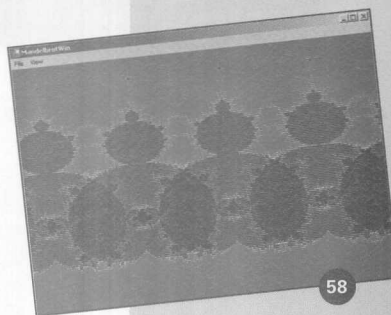
by Don Preuninger and Joe Dour



## 58 C# Explorer Manage C# Objects

Managing objects takes more than conserving memory. Take better advantage of the .NET garbage collector for efficient object management and increased performance.

by Bill Wagner



## 64 Web Services

### Develop Interface-Based .NET Web Services

Separating interface from implementation isn't just for traditional component-oriented apps—its benefits apply to Web Services too. Learn how to develop and consume interface-based Web Services in VS.NET.

by Juval Lowy

www.vbpj.com • www.vcdj.com

To subscribe:

866-387-8275 (international),

850-682-7652 (domestic),

or 800-456-5911

Visual Studio Magazine (ISSN: 1075-1955; Canadian GST: 86831-0764) is published by Fawcett Technical Publications, 209 Hamilton Avenue, Palo Alto, CA, USA, 94301-2500. Tel. 650-833-7100; Fax. 650-853-0230. One year is twelve monthly issues plus two additional special issues at the basic annual subscription rate of \$34.97. Periodicals postage paid at Los Altos, CA and additional mailing offices. Canadian Publication Mail Agreement # 358177. Postmaster: Send address changes to Visual Studio Magazine, P.O. Box 58872, Boulder, CO 80322-8872.

More contents ►

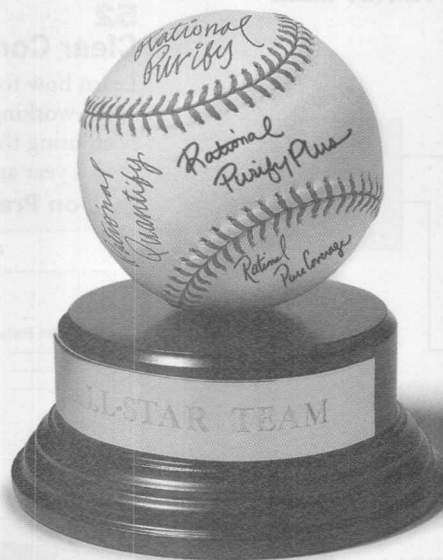


PRINTED IN USA

Content Management integrated with Defect and Change Tracking integrated with Requirements

Software Engineering Best Practices integrated with Visual Modeling integrated with Software Configuration Management integrated with Web

Management integrated with Automated Testing integrated with Software Engineering Best Practices



**It's not often that every player on an all-star team comes from the same team.**

When you're facing software development deadlines, you want a championship team supporting you. Talk about a winning combination. Rational® PurifyPlus brings together the best development tools for Visual Studio — Rational Purify®, Rational Quantify®, and Rational PureCoverage® — to help you develop faster, more reliable code. PurifyPlus combines the power to automatically pinpoint hard-to-find bugs, highlight performance bottlenecks and identify untested code, in a single, inexpensive package that is easy-to-install, easy-to-learn and easy-to-use. And it seamlessly teams up with other Rational tools to provide the industry's only complete software development lifecycle solution. Visit [www.rational.com/offer/allstar3](http://www.rational.com/offer/allstar3) to get the free whitepaper, *Develop Fast, Reliable Code with Rational PurifyPlus*. While you're there, get a free 15 day evaluation copy (downloadable or on CD) and let the PurifyPlus all-star team start working for you today!

**Rational®**  
the e-development company™

©2001 Rational Software Corporation. Rational, the Rational logo, Rational the e-development company, Purify, Quantify, and PureCoverage are trademarks or registered trademarks of Rational Software Corporation in the United States and other countries.

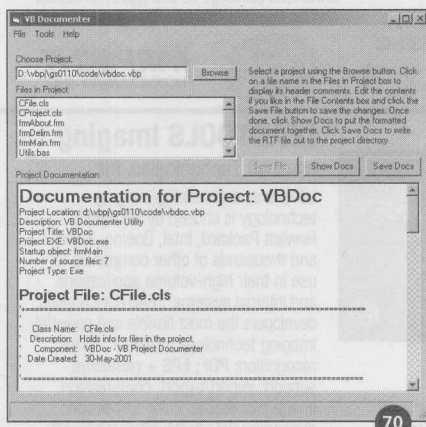


## C O L U M N S

### 70 Getting Started Build a Project Documenter Utility

This ready-made project-documenting utility creates formatted rich text files from any VB6 project. It uses a variety of file-parsing techniques, collections, and classes to gather and format project information.

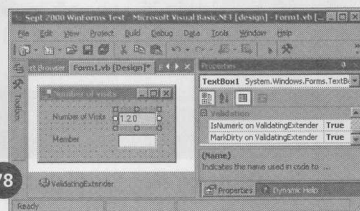
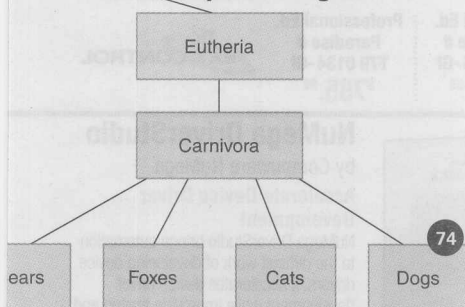
by Stan Schultes



### 74 Best Practices Don't Overuse Inheritance

Understanding animal classification can help you get a handle on inheritance, when to use it, and when other alternatives work better.

by Bill Wagner



### 78 Desktop Developer Whip Forms Into Shape

New in VB.NET, runtime event handlers and property extenders give you consistent event behavior.

by Kathleen Dollard

### 84 Q&A Cure Your Dates of Culture Shock

Our experts show you how to format dates for different regions, use a stack trace to track down bugs, dynamically create ASP .NET server controls that use templates, and strip the time off of a date in SQL Server.

by Jonathan Goodyear  
and Robert Teixeira

### 90 ASP .NET Easy E-Mail—It's About Time

Sending e-mail with ASP .NET is so simplistic, it's almost embarrassing.

by A. Russell Jones

### 94 Database Design Save Application Preferences

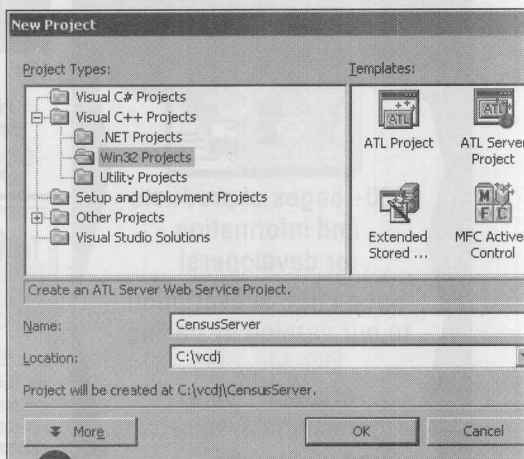
Using classes to save user preferences to your database can work better than using the Registry. You get scalability and centralization ... and you'll make your users happy.

by Dianne Siebold

### 98 Black Belt Develop an ATL Web Service

C# has been praised for many things, including the ability to create Web Services. But you can also create them in VC++.NET and ATL Server.

by Bill Wagner



98

## DEPARTMENTS

### 11 Editor's Note by Patrick Meader

### 13 Reality Check by Elden Nelson

### 14 Letters to the Editor

### 17 First Looks

### 23 Product Listings

### 25 VS All-Stars

### 111 Index of Advertisers

### 112 Guest Opinion by Yasser Shohoud

**Customer Service:** For subscription orders, inquiries, or address changes, call toll-free 866-387-8275 (for domestic subscribers), 850-682-7562 (for international subscribers), or 800-456-5911; send a fax to 650-321-3818; send an e-mail to vbpfawcette@neodata.com or vbpcustomerservice@neodata.com; or write to Visual Studio Magazine, P.O. Box 58872, Boulder, CO 80322-8872. Foreign and Canadian orders must be payable in U.S. dollars plus postage. The surface rate to Canada and Mexico is \$52.97 per year and for all other countries, the air mail rate is \$78.97 per year. Back Issues: To order Visual Studio Magazine back issues, call 650-833-7100 or 800-848-5523 and ask for the customer service department. Back issues cost \$10. Additional postage will be charged to deliveries outside the USA. Reprints: For all Visual Studio Magazine editorial and advertising reprints, please contact PARS International Corporation at 102 West 38th Street, 6th Floor, New York, NY 10018; call 212-221-9595; or fax 212-221-9195. Permissions: If you would like to quote from an article, please fax your request to Fina Pita at 650-853-0230, or mail the request to 209 Hamilton Avenue, Palo Alto, CA 94301-2500, Attn: Fina Pita. Specify the issue date, title of the article, the portion you would like to quote, and the purpose. Photocopy Rights: Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Fina Pita for more information. © Fawcette Technical Publications, all rights reserved. "Visual Studio" is a trademark of Microsoft Corporation and is used by Fawcette Technical Publications under license from the owner. Visual Studio Magazine is an independent publication not affiliated with Microsoft Corporation. Microsoft Corporation is not responsible in any way for the editorial policy or other contents of the publication. All contents of Visual Studio Magazine are copyright © 2001 by Fawcette Technical Publications, unless otherwise noted. "VBITS", "Interactive Developer", "Java Pro", and "Inquiry.com" are trademarks of Fawcette Technical Publications, a California Corporation, James E. Fawcette, President. "Java" is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication.



visit us at **programmersparadise.com**  
or call **800-445-7899**

## FREE Catalog



100+ pages of products  
and information  
for developers!  
Get a FREE subscription  
to our catalog by calling  
**800-445-7899**  
or subscribe at  
**programmersparadise.com**

## Paradise Picks



**New  
Version!**

### Mozquito FML for HomeSite by Mozquito Technologies

**Adding the interactive dimension**  
Mozquito FML for HomeSite brings the strength  
and agility of XHTML-FML to Allaire's easy-to-use  
editor HomeSite. Mozquito Technologies trans-  
formation engine, Mozquito Matrix, translates the  
files into a format all major browsers can display.

#### Benefits:

- Includes 20 new XHTML-FML tags
- Checks your XHTML documents for  
validity and well-formedness
- Enables FML document preview in your  
favorite browsers
- Is a simple-to-use environment
- Contains tags for performing client-side  
calculation and input validation
- Lets you create multiple pages in a  
single document
- Create online forms, surveys, quizzes  
and more with just a few tags.

Paradise #  
**M3A 011B-GI** **\$82.99**

**NASDAQ: PROG**

## SpyWorks Professional

by Desaware

SpyWorks extends the limits of Visual  
Basic by allowing you to do almost  
anything in VB that you can do in other  
languages such as C++. Version 6.3  
includes .NET support for keyboard  
hooks, window hooks and subclassing  
(including cross-task subclassing).  
Examples in both Visual Basic.NET  
and C#. Plus three free updates!



Paradise #  
**D39 0421-GI**  
**\$374.99**

**New  
Version!**



## TestTrack™

by Seapine Software™

TestTrack Pro and Workgroup are the most  
powerful defect tracking solutions for Web,  
Windows, Macintosh, Linux, and Solaris.  
TestTrack Pro 4 delivers a robust solution  
by providing source control integration;  
data field customization; XML and ODBC support; customer and  
user test configurations; e-mail notification; e-mail bug importing;  
duplicate defect handling; release note generation; and unlimited  
customer care. TestTrack Workgroup 2.0 delivers a rich feature set  
for smaller teams.

**New  
Version!**  
4

#### 2.0 Workgroup Ed.

Paradise #  
**S96 0140-GI**  
**\$173.99**

#### 4 Pro Ed.

Paradise #  
**S96 0170-GI**  
**\$210.99**

## Intel® Software Performance Tools

by Intel

**Build Faster Applications, with  
Help Straight from the Experts  
Who Know Processors Best.**



\* Sold separately.  
Call for pricing on compilers.

#### VTune Performance

Analyzer  
Paradise #  
**I23 0153-GI**

**\$517.99**

- VTune™ Performance Analyzer non-  
intrusive sampling and call graph profiling  
offer multiple ways to understand  
code performance.

- Intel® C++ Compiler\* designed from the  
ground up to take full advantage of Intel's  
latest processors.

- Intel® Fortran Compiler\* delivers outstand-  
ing application performance with advanced  
features like Profile-Guided Optimization.



Server  
Paradise #  
**E14 0176-GI**  
**\$228.99**

Workstation  
Paradise #  
**E14 0170-GI**  
**\$48.99**

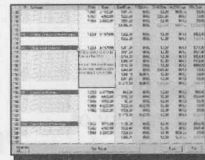
## Diskeeper

Automatic disk defragmenter for Windows 9x/Me/NT/2000

by Executive Software

**NTSL proven to increase system  
performance up to 200%.**

Built to automatically eliminate file fragmenta-  
tion with only a minimum use of system  
resources, Diskeeper represents better  
technology and better economics than any  
other solution available. "Set It and Forget It"®  
functionality gives you totally transparent  
online operation and a lightning fast, Microsoft  
recommended boot-time defrag for MFT and  
Paging Files. Diskeeper provides the fastest  
and most thorough defragmentation possible!



## FarPoint's Spread 3.5—New Version!

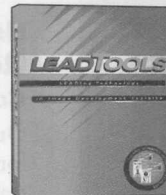
by FarPoint Technologies

Use FarPoint's Spread 3.5 to easily  
incorporate high-level spreadsheet  
and advanced grid features into  
your applications. You get unmatched flexibility at the cell level for  
maximum control over the display and entry of your data, plenty  
of events to respond to user changes and an impressive feature  
list for everything in between, including: import/export capabilities;  
enhanced printing; 12 cell types, formulas;  
and an improved Spread Designer.

Paradise #  
**F02 0133-GI**  
**\$331.99**

**New  
Version!**  
3.5

**FARPOINT  
TECHNOLOGIES INC**



## LEADTOOLS Imaging

by LEAD Technologies, Inc.

LEAD's award-winning imaging  
technology is chosen by Microsoft,  
Hewlett Packard, Intel, Boeing, Xerox,  
and thousands of other companies for  
use in their high-volume applications  
and internal systems. LEADTOOLS gives  
developers the most flexible and powerful  
imaging technology available: forms;  
recognition; PDF; EPS + postscript  
support; import/export; compression;  
image processing; color conversion;  
display/special effects; Internet/intranet;  
annotations; medical imaging; OCR;  
barcode recognition; and much more!

Paradise #  
**L05 0480-GI**  
**\$429.99**

## TX Text Control 9.0

by The Imaging Source

**TX Text Control Does it All.**

A powerful programming component which  
allows developers to easily add into their  
application sophisticated text formatting  
and display capabilities typically seen only  
in large word processing programs.



**New  
Version!**  
9.0

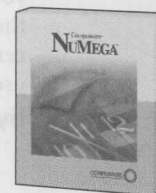
**Download a  
demo today.**

**DEMO  
AVAILABLE**  
programmersparadise.com

Standard Ed.  
Paradise #  
**T79 0136-GI**  
**\$412.99**

Professional Ed.  
Paradise #  
**T79 0134-GI**  
**\$785.99**

**TX  
TEXT CONTROL**



## NuMega DriverStudio

by Compuware NuMega

**Accelerate Device Driver  
Development**

NuMega DriverStudio brings automation  
to the difficult work of developing device  
drivers. It accelerates device driver  
development while improving testing and  
reliability. With its system-wide visibility you  
can easily find and fix Windows driver  
problems on the developer desktop, in the  
testing lab or after deployment.

Paradise #  
**N19 0M33-GI**  
**\$2,499.99**

**Call 800-445-7899 or visit programmersparadise.com**



# Personal Service! Guaranteed Best Prices!



**PR-Tracker™**  
by Softwise Company  
**Bug Tracking...doesn't have to be expensive!**  
Integrate bug tracking over the Internet and a local area network in one database with the same easy-to-use interface.  
PR-Tracker's simple yet powerful interface requires no training and enables you to configure a project in about 15 minutes. Features include: customizable data entry, workflow, views and queries; one click queries; attachments; and email notification.  
**10-24 Users  
Paradise #  
S3R 0142-GI  
\$99.99\***

## PR-Tracker™

by Softwise Company

**Bug Tracking...doesn't have to be expensive!**

Integrate bug tracking over the Internet and a local area network in one database with the same easy-to-use interface.

PR-Tracker's simple yet powerful interface requires no training and enables you to configure a project in about 15 minutes. Features include: customizable data entry, workflow, views and queries; one click queries; attachments; and email notification.

\* Price per user.



**RoboHelp® Enterprise**  
Discover How People are Using Your Application and Help System  
The newest Help authoring tool in the RoboHelp family provides:  
• All the Help authoring benefits of RoboHelp Office, plus...  
• Integrated end user feedback reports to improve the effectiveness of your application and its Help system  
• Natural language search to make searching easier  
• Project merging to streamline team development.  
**Paradise #  
E75 0290-GI  
\$1,588.99\***

## RoboHelp® Enterprise

Discover How People are Using Your Application and Help System  
The newest Help authoring tool in the RoboHelp family provides:

- All the Help authoring benefits of RoboHelp Office, plus...
- Integrated end user feedback reports to improve the effectiveness of your application and its Help system
- Natural language search to make searching easier
- Project merging to streamline team development.

\* Limited time pricing.



## GUARANTEED BEST PRICES\*

Should you see one of these products listed at a lower price in another ad in this magazine, CALL US! **We'll beat the price**, and still offer our same quality service and support!

\*Terms of the offer:

- Offer good through October 31, 2001
- Applicable to pricing on current versions of software listed
- October issue prices only
- Offer does not apply towards obvious errors in competitors' ads
- Subject to same terms and conditions

Prices subject to change.  
Not responsible for typographical errors.



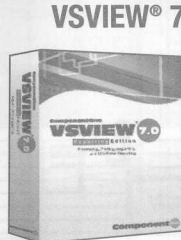
**DevTrack**  
by TechExcel  
**Powerful Defect Tracking**  
DevTrack is the premier defect- and project-tracking tool for software development teams, helping to ensure that development projects finish on time and on budget. DevTrack comprehensively tracks and manages all defects, change/feature requests, and all other development issues. DevTrack also provides powerful workflow and process automation features, robust searching and reporting, and comprehensive point-and-click customization. Intuitive and powerful, DevTrack provides a scalable out-of-the-box solution, at a great value.  
**Paradise #  
T34 0170-GI  
\$463.99**

## DevTrack

by TechExcel

**Powerful Defect Tracking**

DevTrack is the premier defect- and project-tracking tool for software development teams, helping to ensure that development projects finish on time and on budget. DevTrack comprehensively tracks and manages all defects, change/feature requests, and all other development issues. DevTrack also provides powerful workflow and process automation features, robust searching and reporting, and comprehensive point-and-click customization. Intuitive and powerful, DevTrack provides a scalable out-of-the-box solution, at a great value.

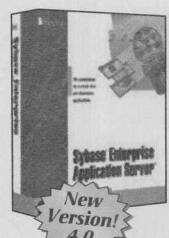


**VSVIEW® 7.0 Reporting Edition**  
by ComponentOne  
ComponentOne VSVIEW Reporting Edition adds powerful, flexible database reporting to one of our most popular ActiveX® controls, VSVIEW Classic Edition. Now you can quickly and easily create Microsoft® Access-style database reports for your Visual Basic, VC++, and C++Builder applications, plus preview, print, and export those reports on your computer or over the Web.  
**Paradise #  
C18 0350-GI  
\$476.99**

## VSVIEW® 7.0 Reporting Edition

by ComponentOne

ComponentOne VSVIEW Reporting Edition adds powerful, flexible database reporting to one of our most popular ActiveX® controls, VSVIEW Classic Edition. Now you can quickly and easily create Microsoft® Access-style database reports for your Visual Basic, VC++, and C++Builder applications, plus preview, print, and export those reports on your computer or over the Web.



**Sybase EAServer 4.0**  
by Sybase  
Using leading J2EE 1.3 technology, Sybase EAServer 4.0 is the high performance, scalable e-Business engine that can give you the competitive edge to succeed in today's e-Business environment. Sybase EAServer is the premier choice for application development because of its reliable, dependable, and open architecture. This engineering design enables Sybase EAServer to adapt to existing IT architectures to support scalable Web applications.  
**Paradise #  
S84 096W-GI  
\$5,477.99**

## Sybase EAServer 4.0

by Sybase

Using leading J2EE 1.3 technology, Sybase EAServer 4.0 is the high performance, scalable e-Business engine that can give you the competitive edge to succeed in today's e-Business environment. Sybase EAServer is the premier choice for application development because of its reliable, dependable, and open architecture. This engineering design enables Sybase EAServer to adapt to existing IT architectures to support scalable Web applications.



**Visual Build**  
by Kinook Software  
Automate Software Builds  
Visual Build empowers developers to save time and money by creating an automated, repeatable process for building software. It is a powerful but affordable solution that will automate the build process without a huge commitment of time, and it works with the tools you're using today.  
**Paradise #  
K21 0115-GI  
\$148.99**

## Visual Build

by Kinook Software

**Automate Software Builds**

Visual Build empowers developers to save time and money by creating an automated, repeatable process for building software. It is a powerful but affordable solution that will automate the build process without a huge commitment of time, and it works with the tools you're using today.

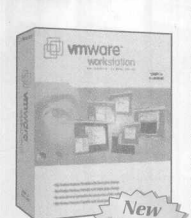


**OLETools 7.0**  
by Becubed Software  
Build professional front-end applications with this collection of components. OLETools is a complete set of 60 32-bit ActiveX components including: a Tab control—for tabbed interfaces; a Tree control—for displaying hierarchical data; multi-column List and Combo Boxes; and various masked input controls for implementing advanced data verification. Also, includes a 3D calendar, and multimedia components for playing back AVI, and WAV files from within your application.  
**Paradise #  
B30 0731-GI  
\$287.99**

## OLETools

by Becubed

Build professional front-end applications with this collection of components. OLETools is a complete set of 60 32-bit ActiveX components including: a Tab control—for tabbed interfaces; a Tree control—for displaying hierarchical data; multi-column List and Combo Boxes; and various masked input controls for implementing advanced data verification. Also, includes a 3D calendar, and multimedia components for playing back AVI, and WAV files from within your application.



**VMware Workstation**  
VMware Workstation increases the productivity of developers and other technical professionals by enabling them to run multiple operating systems in secure, transportable, high-performance virtual computers on physical computers. As a result, our customers spend more time delivering tangible value to their businesses and less time installing operating systems, rebooting or reconfiguring hardware. VMware Workstation supports standard operating systems such as Microsoft Windows NT, Windows 2000, Windows XP and Linux.  
**Paradise #  
V55 011M-GI  
\$325.99**

## VMware Workstation

VMware Workstation increases the productivity of developers and other technical professionals by enabling them to run multiple operating systems in secure, transportable, high-performance virtual computers on physical computers. As a result, our customers spend more time delivering tangible value to their businesses and less time installing operating systems, rebooting or reconfiguring hardware. VMware Workstation supports standard operating systems such as Microsoft Windows NT, Windows 2000, Windows XP and Linux.



## Visual SlickEdit® v6.0

by MicroEdge, Inc.

Source code analysis, side-by-side file comparison, extensive language support, easy to implement and configurable to your coding styles. New Java capabilities including Java Tool Integration for setting compile, execute, debug, javadoc and jar options.

**Paradise #  
M39 0151-GI**

**\$268.99**



## TeeChart Pro ActiveX Version 5

by Steema Software

The hottest in COM Charting for Windows and the Web! Extensive example code for ASP/IIS, Visual Studio, Office and more. Multiple export formats, Multiple axes, Multiple Chart types in 2D, 3D and OpenGL. Custom Canvas, Grid, Toolbar, built-in ADO and .NET support and many specialist financial functions including MACD, ADX and Stochastic.

Download a demo today.



**Paradise #  
SOT 0113 E-WP\***

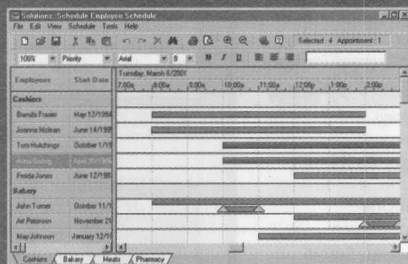
**\$284.99**

**In Canada—call 1-888-423-2700 (www.pparadise.on.ca)**



# Imagine what you can create with the right **components**.

DBI has been helping the world's top developers build better apps for years. With DBI scheduling components you'll exceed all expectations... even your own.



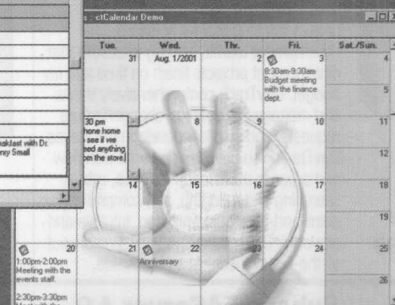
## Solutions::Schedule

Schedule up to 32,000 entities at a time and present the schedule in increments of hours, days or weeks. Schedule multiple lines for each entity with a wide variety of time-bar styles. Drag and drop time-bars within the schedule. The ideal solution for employee and resource scheduling.



## Solutions::PIM Professional

Seventeen personal information management components designed to work together. Multi-column day view, week view, calendars, alarms, date and time edit control — all use common properties and events for seamless integration into any application.



## Calendar Tools

Five fast, easy to use drop-in calendar components including monthly and yearly calendars, drop date edit control, scrolling calendar and more. Calendar Tools is an affordable, functional component solution to meet your design needs.

**DBI components.**  
**Built for Visual Studio™**  
**ready for the future.**



## put it together

Download a free trial version: [www.dbi-tech.com](http://www.dbi-tech.com)  
or call toll free: 1-800-670-8045.

# Feedback Shapes Content

*Letters, we get letters. We get stacks and stacks of letters.* —“The CBS Mailbag Song”

It's been a busy month for e-mail and newsgroup postings—some positive, some negative, a lot wait-and-see. Of course, that wasn't unexpected in light of the magazine's recent name change.

The reaction to the name change has largely mirrored the reaction of developers to Visual Studio .NET. Those who think Microsoft is moving in the right direction with .NET and are excited about the capabilities of that tool have generally welcomed the changes to the magazine. Those who think Microsoft has made a misstep with VS.NET generally feel the magazine is exacerbating Microsoft's misstep with one of its own.

I thought I'd share some of this feedback, both to invite more and to provide a window into what kinds of changes your fellow readers are requesting. I'm also hoping others will use this as an opportunity to tell us what kind of magazine they want to see. I've chosen a handful of posts from the Talk to the Editors forum—posts that are representative of the range of opinions we've received on the subject.

The first e-mail comes from Michael Welch, who responded positively to the change: “I think the change is great. One magazine that covers many of the topics I'm interested in sounds great. I don't want five different magazines covering VB, C#, SQL, XML, ADO, the .NET Framework, and so on. Having one magazine that not only addresses several technologies, but some of the best ways to integrate them, sounds fantastic. Truthfully, I don't see how effective it would be to keep them all separate for much longer.”

Larry Linson weighed in at the other end of the spectrum: “I tolerated the ‘other subjects’ introduced into *Visual Basic Programmer's Journal* after it apparently decided that ‘enterprise solutions’ were the *only* thing that mattered and everything else was just Getting Started or ‘101’—beginner stuff. I tolerated that because there was still enough, barely, of the level of VB that I use that would be helpful to me. Further diluting the content with everything VS, especially with an emphasis on .NET (which, AFAIK, is only in beta still), cannot help but reduce the coverage of interest to me below the level warranting a subscription.”

Dan Fergus took a more wait-and-see approach. Responding to a poster who requested we expand the number of pages in *Visual Studio Magazine*, Dan wrote, “I don't think they are

going to triple the page count. If some of the articles are Studio or namespace related, they can cover both VB and C# programmers and still be useful. Maybe one-third VB, one-third C#, and one-third Visual Studio topics. I could live with that! I'll have to wait and see how it falls out before I call Patrick and yell at him!”

The prospect of being yelled at notwithstanding, I appreciate the feedback offered so far. I wish everyone felt as Michael Welch does, but I would settle for everyone to feel as Dan Fergus does—to look at the new magazine carefully, then let us know what you like and don't like. For those who feel as Larry Linson does, let us know which kinds of articles you want to read about in the magazine—it's the surest way to find them and the mix you want in the future. I'd also like to reiterate that over time, we'll adjust the mix so it's representative of what you use Visual Studio for—in other words, your feedback is important.

The hardest part of this job is figuring out what readers want. Readers who write in and tell you what they want to read do the hard part for you. In such cases, I occasionally follow the advice of filmmaker Jean-Luc Godard, who once said the best way to critique a movie is to make another movie. In other words, it can be an opportunity to invite a critic to write the article he or she wants to read.

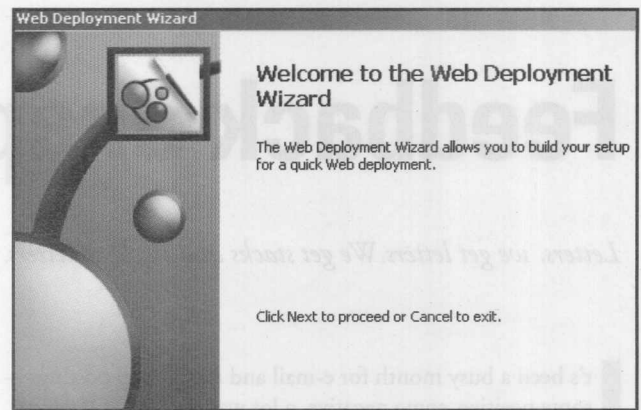
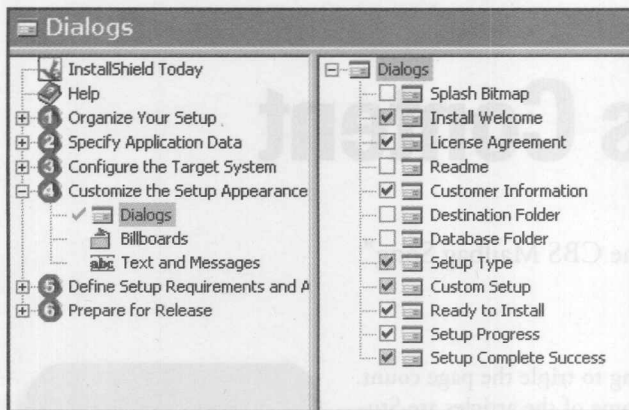
Of course, you don't have to write an article if you want to see a particular subject appear in the magazine. Instead, you can visit our research site at [www.ftpresearch.com/vsmag](http://www.ftpresearch.com/vsmag), which we'll update monthly as each issue ships. This month, one random respondent will win a Palm V. If you don't like to fill out surveys but want to share your thoughts, drop me an e-mail at [ednote@fawcette.com](mailto:ednote@fawcette.com) or e-mail VSM's editors directly at [vsmedit@fawcette.com](mailto:vsmedit@fawcette.com). **VSM**



**Patrick Meader**  
Editor in Chief

What do you want to read about? Tell me at [ednote@fawcette.com](mailto:ednote@fawcette.com), or the magazine's editors at [vsmedit@fawcette.com](mailto:vsmedit@fawcette.com).

# Quick. Easy. InstallShield Express.



## Industry-Standard Installations out of the Box in Minutes.

You need quick installation development. And that means easy-to-use. But you won't sacrifice quality to get it. No sweat. Your answer is InstallShield Express.

InstallShield Express is the easiest and most economical solution for quickly developing straightforward setups. An intuitive checklist guides development. You just point and click through one section, check it off, and move on to the next. In just six quick steps, you complete everything necessary to build your installation.

Remember: a robust application doesn't necessarily mean a complex setup. If you need to copy files, make registry edits, install third-party run-time executables, or even deploy on the Web, InstallShield

Express can have you covered in no time, getting software to your users faster—even beating your competition.

And now InstallShield Express features One-Click Install™ Web deployment technology. Your users get one seamless experience right through download and installation, skipping "save as" dialogs and avoiding endless file searching. And you get to digitally sign and password-protect your setups.

Plus One-Click Install features let users avoid downloading several setup components if they already exist on their PCs—speeding things even more. Your customers will use your software sooner, have fewer hassles, and develop an even

better opinion of your technology. Speedy development. And speedy Web delivery. All with InstallShield Express.

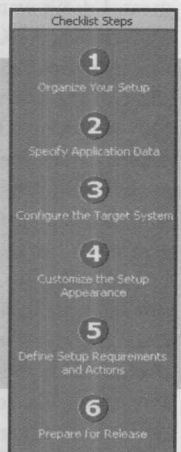
So find out more. An evaluation copy of InstallShield Express is available from our Web site. But with such value available at so tempting a price, why not order today? With our 30-day money-back guarantee, you can get your hands on the full product risk-free and right away. And of course the award-winning InstallShield support team is available for help on any challenge.

So there you have it. Ease of use. The security of the industry standard. And most importantly, the speed you crave. So rush to [www.installshield.com](http://www.installshield.com) and find out more right now!

## key features

Bottom line: it's an InstallShield setup. So everyone knows it works and works well. Here's a few of the features that make it happen for InstallShield Express.

- Enhanced Checklist Design Environment
- Visual Basic Project Scanner
- Static and Dynamic File Dependency Scanners
- VBScript, DLL, and EXE Custom Actions
- One-Click Install™ Web Deployment
- 14 World-Wide Languages
- Advanced Compression Option
- Nearly 50 Third-Party Technologies
- Upgrade Authoring
- Dialog Text Editor
- Windows XP Support
- Upgrade to InstallShield Developer



GET YOUR FULL-FEATURED EVALUATION  
AT [WWW.INSTALLSHIELD.COM](http://WWW.INSTALLSHIELD.COM)



© 2001 InstallShield Software Corporation. All rights reserved. InstallShield is a registered trademark and service mark of InstallShield Software Corporation. All other trademarks are the property of their respective owners.

**InstallShield**  
SOFTWARE CORPORATION



# Pop Quiz

**Y**ou're a developer. Therefore, you have a nasally voice, a pocket protector, a complete inability to relate with people, an almost supernatural gift for breaking through security measures (coupled, by the way, with a disregard for others' privacy), and an intense love of math.

What? That's all (or at least mostly) wrong? Of course it is. I'm not exactly sure where the programmer-as-hopeless-geek-outcast stereotype came from (*Revenge of the Nerds* had a lot to do with it, is my guess), but I *do* have a hunch that—like other professionals—we share a number of preferences and experiences that give us a certain general character. That character just happens to be a lot different than what most people think.

So, here are some questions you can ask yourself, then ask other developers you know. See if you fit the same mold. E-mail me ([realitycheck@fawcette.com](mailto:realitycheck@fawcette.com)) with your answers and conclusions. As an incentive, I'll send a PlayStation 2—I haven't been able to get my hands on an Xbox yet—to someone at random. I'm interested to see how we're the same ... and different.

**How did you get started?** What was the first program (apart from "Hello World") you ever wrote, approximately how many lines of code did it require, and what language did you use? *Bonus:* Do you remember the first time you sat at a desktop computer? What computer was it, and when? *Secret Bonus:* Did you ever use punchcards? If so, how did you stay sane?

**Why are you here?** Do you remember making a conscious decision to be a developer, or did you just sort of slide into the role? If the former, describe what made you want to be a programmer. If the latter, explain the (probably bizarre) set of circumstances leading you to where you are today.

**How important is college?** Do you have a formal computer-science-related degree? If not, do you envy those who do? If you do, do you feel your education has made you a better programmer? *Bonus:* Is there a point of diminishing returns in college-level programming instruction? For example, does having a doctorate in computer science mean you're a guru, or does it mean you've been out of the trenches too long?

**What do you like?** Which part of the development process do you like best? Designing the program? Writing new code? Troubleshooting? Streamlining existing apps? What do you like about it? *Bonus:* Which part do you hate most?

**How much does money matter?** Are you programming more because you love to program or because there's good money in it? Would you still be a programmer if it paid, say, about the same as being an elementary school teacher?

**How do you explain yourself?** You're talking with somebody who clearly doesn't have the foggiest notion of what a program is, much less what a programmer does. This person wants to know what you do for a living. Do you get extremely vague ("Oh, I work with computers ..."), or do you settle in and try to make him or her understand? *Bonus:* Recount, in detail, how you made (or failed to make) this person understand.

**Do you like math?** Everyone thinks that because you're a programmer, you must be a math wiz. Are you?

**What would you have done for a living 200 years ago?** The first floppy disk won't be on store shelves 'til loooooong after you're dead. What do you do instead?

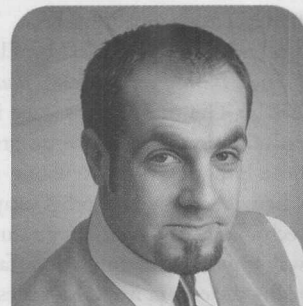
**Have you ever hacked?** Have you ever used your programming skills to do something you shouldn't have? If so, do you now regret it? If not, are you sometimes tempted? *Bonus:* One of the most common things you see computer programmers do in movies is break codes. Could you—right this instant, without doing a bunch of research—do this, if you wanted to?

**What's your proudest programming moment?** At some point, you wrote a program—or part of a program—that proves you're a genius. Did anybody else recognize your Einstein moment? *Bonus:* Did you go out of your way to make *sure* somebody recognized your development epiphany?

**What's your biggest moment of programming shame?** At some point, you (probably) wrote something you now realize is complete trash. Did you realize it was garbage at the time? Did you fess up? Does it still haunt you at night?

**What's the best trade show swag you've ever received?** Surely you've admitted to yourself that one of the reasons you go to trade shows is to pick up the free stuff. What's the coolest free thing you've ever gotten?

**What would you say to set the record straight?** A major network calls you and says they've reserved two minutes of air-time for you to explain to the world what developers are really like. What do you say? **VSM**



**Elden Nelson**  
Editor

How are you the same as—and different than—other developers? Write to me at [realitycheck@fawcette.com](mailto:realitycheck@fawcette.com).

Letters to *Visual Studio Magazine* are welcome. Letters must include your name, address, and daytime phone number to be considered for publication. Letters might be edited for form, fit, and style. Please send them to Letters to the Editor, c/o *Visual Studio Magazine*, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax them to 650-853-0230; or e-mail them to [vsmedit@fawcette.com](mailto:vsmedit@fawcette.com).

## Newer ≠ Better

In regards to Reality Check, "Are You a Techno-Retro-Grouch?" by Elden Nelson [*Visual Basic Programmer's Journal* August 2001]: The problem, as I see it, is one of attitude. I think a lot of programmers believe programming is about technology, when it's really the other way around—technology is about programming. It's a way to get certain things done faster and not really the end in itself. Libraries and frameworks and platforms will never be a substitute for creativity and perseverance. People seem more interested in acquiring a long list of acronyms (XML, JSP, EJB ...) rather than acquiring a deep understanding of the means of creating algorithms and machine architecture. Without knowledge of these two things, it's difficult to do much more than assemble some configuration of a prefabricated machine.

I see a lot of new applications but few new concepts. Maybe I am a techno-retro-grouch, but to be one at the age of 26 seems a sad commentary. It's not that I want to go back to hand-crafting 3-D graphics algorithms in 8088 assembler or that I believe you're not a programmer if you don't code in C. I'm just a little saddened when I meet programmers who cannot appreciate the beauty of a game such as Pong.

*Marvin Wilson, New York*

I became tired of the incessant and increasing "progress" in technology around the time Windows debuted. Every few months, there's a new operating system and a new language version. We learn HTML and out comes XML. We learn VB6 and out comes VB.NET. We learn Access and they want Oracle. We have no time to devote to applications due to system software updates.

I don't see how anyone has the time (or ability) to keep up with this nonsense. Recently, *VBPI* had an article on VB.NET and its differences from VB6 ["Drill Down on VB.NET" by James Foxall, February 2001].

Our current code won't work. I sighed when I learned about the demise of control arrays and considered the redundant code necessary to replace them. And then, the bombshell. No native file support! WHAT? I can't access a simple flat file? No byte-stream I/O? That means I can't use VB.NET. It also means I *won't* use it.

I have had enough. I'm getting out. I retired last month, and shall use VB6 at home until it rots, at which time I shall revert to paper and pen.

*Dave Westfall, Southfield, Mich.*

## VB Does Graphics, Too

Thank you so much for the article "Take Control of DirectX" by Jonny Anderson [*Visual Basic Programmer's Journal* June 2001]. In the past, so much of your magazine was devoted to databases, it really was a breath of fresh air to see an article on graphics. VB is capable of so much more. My hat is off to you.

*Robert Martin, Longview, Texas*

## Get the Code

Use these Locator+ codes to download the code for this issue of *VSM* at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com).



**VS0110:** All the listings and code files for the October 2001 issue of *VSM* in one ZIP file

**VS0110RT:** "VB.NET: Pointers Add Power and Safety": Listing 1, which shows you how to create a read-only view into a text file

**VS0110DT:** "Enhance Performance With ADO Connection Objects": the code to create the COM and MTS implementations of the DLL described in the article

**VS0110CE:** C# Explorer: Listings 1 through 4 and the finished project that creates, displays, and animates the Mandelbrot set

**VS0110WS:** Web Services: a basic, non-interface-based Web Service; two interface-based Web Services; a test client that uses them all; and Listings 4 and 5

**VS0110GS:** Getting Started: a complete utility to create VB project documentation, as well as Listings A, B, and C, which weren't printed due to space constraints

**VS0110DT:** Desktop Developer: the extender class and a test project that demonstrates its use

**VS0110QA:** Q&A: C# code for the first two questions (Listings A and B), VB.NET and C# ASPX files for the dynamically created Repeater control, and the SQL script for stripping the time off of a date in SQL Server

**VS0110AN:** ASP.NET: code to create a WebForm that users fill out to send e-mail from your Web site

**VS0110DD:** Database Design: a sample application demonstrating the classes for saving and retrieving user preferences

**VS0110BB:** Black Belt: a Web Service built using C++ and ATL Server, that same Web Service built using C#, and a C# client that uses the Web Service

## GROUP PUBLISHER

Jeff Hadfield

## VSM STAFF EDITORS

Patrick Meader, *Editor in Chief*

Elden Nelson, *Editor*

Nicole Pizzurro, *Managing Editor*

Nick Fuentes, *Associate Managing Editor*

Nina Goldschlager, *Executive Editor*

Susannah Pfalzer, *Senior Editor*

## CONTRIBUTING EDITORS

Daniel Appleman, Francesco Balena, Chris Barlow, Andrew J. Brust

Matthew Curland, Dan Fergus, Keith Franklin, Joel Garcia

Jonathan Goodyear, Mary V. Hooke, Roger Jennings, L.J. Johnson

A. Russell Jones, Don Kiely, Chris Kinsman, Deborah Kurata

Bill McCarthy, Karl E. Peterson, Keith Pleas, Stan Schultes

Ron Schwarz, Yasser Shohoud, Dianne Siebold, Phil Weber, Jonathan Zuck

## TECHNICAL REVIEWERS

Ash Rofail, *Senior Technical Reviewer*

Mary V. Hooke, Eric Kinatader, Robert Lair, Brian Noyes

Ken Rabold, Ash Rofail, Ron Schwarz, Bill Wagner

## EDITORIAL ADVISORY BOARD

Francesco Balena, Richard Finkelstein, Craig Goren, Jim Lisiak

Bill Neville, Karl E. Peterson, Joe Robison, William Storage

## ART & PRODUCTION

Michael Hollister, *Vice President, Art & Production*

Bruce Gardner, *Senior Art Director*

Casey Mallough, *Associate Art Director*

Christine Marquez, *Production Director*

Michael Furukawa, *Senior Production Manager*

Phil Weber, *Software Developer*

Sasha Owen, *Senior Producer*

Mona Bedi, *Associate Web Producer*

Shane Lee, *Associate Web Producer*

Mary Whalen, *Associate Web Producer*

Hao Luu, *Web Associate*

Sheila Quirke, *Production Artist*

## ADVERTISING SALES

Karin Becker, *Group Advertising Director*

Brigit Chennells, *Regional Account Manager (Southwest)*

Mark Romasco, *Regional Account Manager (Northeast)*

Tara Stafford, *Regional Account Manager (Northwest)*

David Seymour, *Inside Sales Manager*

Fina Pita, *Production Coordinator*

## CIRCULATION

Meg Clark, *Circulation Manager*

Gerry Guzman, *Circulation Associate*

Joyce Hoelzle, *Circulation Associate*

## MARKETING

Rob Tobias, *Senior Vice President, Marketing*

Libby Dickinson, *Marketing Communications Manager*

Margaret Horoszko, *Senior Marketing Designer*

Ray Long, *Marketing Production Coordinator*

Susan Ogren, *Marketing Specialist*

## CONFERENCES

Tena Carter, *Vice President, Conferences*

Tim Smith, *General Manager, Conferences*

Tanya Cecchin, *Sales Manager, West*

Heidi Corcoran, *Conference Sales Planner*

Lee Thé, *Windows Editorial Director*

Leslie Hasvold, *Conference Operations Planner*

Thea Schwartz, *Conference Marketing Director*

Staci Matlack, *Conference Planner*

Jennifer Brucker, *Marketing Manager*

Nicole Simonich, *Marketing Planner*

Shannon Sullivan, *Marketing Planner*

Maryam Ghaemmaghami, *Associate Editor*

Sarah Lemonds, *Marketing Editorial Assistant*

## CUSTOMER SERVICE

Melanie Meinert, *Customer Service Representative*

Laurie Groenewold, *Customer Service Representative*

Kelsi McConville, *Customer Service Representative*

Nicholas Cadigan, *Customer Service Representative*

## OPERATIONS

Darlyn Phillips, *Director of Finance*

Linda Brooks, *Assistant Controller*

Neema Buruku, *Financial Analyst*

Elena Ostrovsky, *Accounts Payable Accountant*

Iain Neillands, *Accounts Receivable Accountant*

Lina Yonemitsu, *Collections Analyst/Cash Management Consultant*

Nancy Burgoa, *Accountant*

Betty Wu, *Staff Accountant*

Richard Feierstein, *Payroll Accountant*

Philip Ziman, *HR and Facilities Director*

Pamela Davis, *HR Manager*

Nancy Wilson, *Office Administration/Facilities Manager*

Tin Cao, *System Administrator*

Wendy Gray, *Receptionist*

## FAWCETTE TECHNICAL PUBLICATIONS

James E. Fawcette, *President*

Patricia Cameron-Loyd, *Chief Operations Officer*

John Sutton, *Executive Vice President and Chief Financial Officer*

Matt Carter, *Vice President, Internet Products*

Aaron Weule, *Vice President of Technology and Chief Information Officer*

Paul Spyskma, *Publisher, FTP Advertising Network*

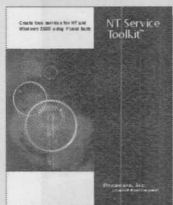
Wilson, Sonsini, Goodrich & Rosati, *Corporate Counsel*



# The .NET Learning Curve

## Dan Appleman, leading author on VB

The biggest challenge .NET poses to Windows developers today is not deployment - it's education. The .NET Framework is so huge and so different, that you'll need to invest significant time studying and experimenting with .NET before you start deploying real applications. How does one go about learning .NET? Expensive courses and conferences? Books? Articles? Regardless, it is essential to first learn the core concepts of .NET. That is the foundation on which you can base further learning, and that's the focus of my new book "**Moving to VB.NET: Strategies, Concepts and Code**". You can read the introduction and sample text at [www.desaware.com](http://www.desaware.com). There you'll also read about SpyWorks 6.3 with .NET support, VersionStamper's support for .NET applications that use COM components, and new products designed specifically for .NET. .... Dan



### NT Service Toolkit

#### Full featured services with VB6!

Visual Basic programmers have long searched for a good way to create NT Services using Visual Basic. Yet every solution they found had serious limitations and compromises - until now. Desaware's NT Service Toolkit provides all the power of a C++ service from VB - but the

services are far easier to create and debug!

- **Debug your service using the VB environment** - while it is running as a service!
- Automatic **background threads** for waiting on NT synchronization objects.
- Access and control services through **COM or DCOM**.
- Expose client objects via **COM or DCOM**.
- Client objects run on a **thread pool for scalability**.
- Create background threads for asynchronous operations.
- Create control panel applets to manage your services.
- All service notifications (including those new to Win 2000).
- Simulator allows testing without installing as a service.
- Impersonation allows acting on behalf of clients.
- Access to all service configuration parameters.



### Escape DLL Hell!

**VersionStamper** helps eliminate incompatibility problems that can occur when distributing component based applications - **including .NET applications that use COM components**. Your application can analyze a client system using an embedded dependency list or data from your Web or FTP site. Problems can be resolved by

notifying the user, reporting via Email, or **automatic download** of the correct components from your Web or FTP site.



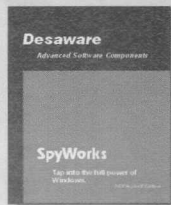
### New! Event Log Toolkit

Create custom event sources with this easy to use toolkit. Eliminates the problem with VB6 in which all events are logged as coming from VBRuntime. Event sources are easy to distribute, self-installing, and support multiple languages. Included code demonstrates advanced event reporting API calls.



### Data Storage for VB, VBA and Internet Applications

**StorageTools** allows you to create complex documents, each of which can be subdivided into multiple named blocks of data. **StorageTools** actually gives you **more data storage flexibility than most databases, without the overhead!**



### SpyWorks® 6.3 - The ultimate low level toolkit, now compatible with VB.NET and C# for VS.NET Beta 2

**Advanced Subclassing and Hooks** - In-and cross process subclassing for VB6 and .NET! Use **Windows Hooks** to intercept messages and keys for a window, process or the entire system. **Create Background threads** for your

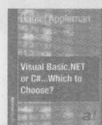
VB6 DLL COM objects. Create true **Function Exports** from your VB6 DLLs. Handle Internet/Intranet communications with our **Winsock Component** - with VB source. Create **Control Panel Applets**. **Implement or call ANY interface** - No type library. *And much more....*

## Books



### Moving to VB.Net: Strategies, Concepts and Code

VB.Net is not Visual Basic. Porting is stupid. COM is "dead". These are just a few of the things you'll learn as Dan takes you on a journey unlike any other into the world of VB.Net. Covers adoption strategies, unlearning VB6 concepts that are fatal in VB.Net, and analysis of language changes that goes beyond the documentation. APRESS, ISBN 1-893115-97-6



**Visual Basic.NET or C# - Which to Choose?** - A new PDF based e-book comparing VB.NET and C# available now at [www.desaware.com/VBorCsharp.htm](http://www.desaware.com/VBorCsharp.htm).



**How Computer Programming Works** - The ideal book for beginning programmers. Teaches the fundamentals of programming using illustrations and real world analogies. APRESS, ISBN 1-893115-23-2

### Also available

**Dan Appleman's Visual Basic Programmer's Guide to the Win32 API** - The core Windows Software Development Kit rewritten for VB programmers. **Dan Appleman's Win32 API Puzzle Book and Tutorial for VB Programmers** teaches you to call any API function from VB5 & 6. **Developing COM/ActiveX Components with Visual Basic 6.0**. The essentials of COM and component development for VB.

For more information, please visit...

**[www.desaware.com](http://www.desaware.com)**

### Desaware Inc.

1100 E. Hamilton Ave., Suite 4  
Campbell, CA 95008  
tel: (408) 377-4770 fax: (408) 371-3530  
email: [support@desaware.com](mailto:support@desaware.com)





## Herd about COW?

### Caché Objects for the Web

Introducing the **fastest** way to build sophisticated

database applications for the Web. Featuring dynamic server pages, and simple-to-use yet powerful object technology. With the world's fastest and most scalable Web database, Caché, already built in. Compatible with other popular databases and leading Web design tools.



Download a fully-functional COW, or request it on CD, free at: [www.CACHE-COW.com](http://www.CACHE-COW.com)

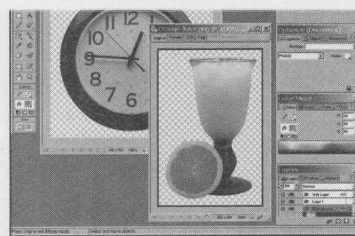
Or call 1-800-753-2571

# A Web Site Dream Team

The technologies needed to create increasingly sophisticated Web applications are growing more complex. Gone are the days when building a Web site with Notepad was feasible. If you need to put together a site—in Internet time—you need the right tools.

I've long relied on Microsoft's FrontPage and Visual InterDev, but Macromedia's Dreamweaver 4 Fireworks 4 Suite (bundling the company's two products together and giving it way too long of a name) provides some tempting reasons to switch. This is a comprehensive package of Web and graphics development tools that has the added benefit of extensibility.

Dreamweaver is a Web site design and development tool that includes an HTML editor, support for W3C standards and accessibility features, support for CSS, and, well, everything you need to



**Graphics to Die For.** Fireworks 4, which comprises half the suite, provides a rich set of bitmap and vector tools that let you create, edit, and animate your Web graphics. Macromedia is positioning Fireworks 4 as a direct competitor to Adobe Photoshop 6 for professional Web design.

build rich Web sites and applications. This latest version adds a slew of new features, including an integrated text editor with color-coded HTML and scripting code, a JavaScript debugger, and an integrated O'Reilly code reference for information about tags and objects. One cool new feature is the Asset Management Tool, which lets you track all site media—including site images, colors, and external URLs—in a central location.

The other product in the suite is Dreamweaver 4, which you use to design and optimize Web graphics for easy integration into your site. It also sports many new features, such as a popup menu creator to implement cool site navigation without writing all the messy client-side code; a Layers panel, which has expanded thumbnails of each layer's objects and advanced masking capabilities; and a Batch Processing Wizard for executing multiple processes. Fireworks' support for importing and/or exporting graphic elements now includes Photoshop, FreeHand 9, EPS, and wireless BMPs.

## Dreamweaver 4 Fireworks 4 Studio

### Macromedia

**Web:** [www.macromedia.com](http://www.macromedia.com) **Phone:** 415-252-2000

**Price:** \$399

**Quick Facts:** Extensive collection of Web site and application tools, with plenty of support for graphics-intensive sites.

**Pros:** Virtually everything is in the box; great documentation and support; many new features to round out the products.

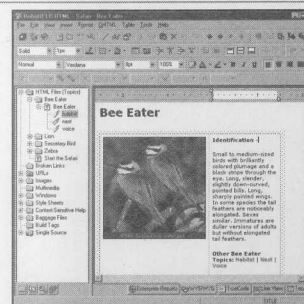
**Cons:** Time-consuming to learn; lack of good programming database support (available in the UltraDev edition).

# Help is on the Way

One thing that seems to pervade every type of project, regardless of language or the degree of programmer proficiency, is a lack of documentation—both written manuals and help files.

RoboHelp Enterprise is a one-size-fits-all attempt to address that situation. After spending some time with it, I've concluded that it's a real powerhouse. The Enterprise incarnation is a superset of the other target-specific RoboHelp versions. It also adds server-side functionality not present in any other version.

These additional features use eHelp's MindReader server, which runs under Internet Information Server (IIS). There are two types of MindReader features—a natural-language search engine, which is aimed at your end users, and a set of report functions, aimed at the administrative and development people working on the project. The natural-language feature requires the RoboHelp Enterprise project type; it won't work with HTML Help, WinHelp, or any of the other targeted platforms that RoboHelp supports. This isn't an issue for an



**An IDE by Any Other Name.** RoboHelp Enterprise's Authoring Environment has a similar feel to FrontPage, and the resemblance is more than skin-deep—you can use it to develop Web sites, too, complete with natural-language search functions. The HTML Help Environment handles all the help formats *except* for legacy WinHelp HLP projects; the RoboHelp Classic Environment, included with the Enterprise edition, manages these projects.

organizational intranet or even an Internet application that can access your Web server, but it precludes you from adding natural-language searches to your standalone, non-networked projects.

The Enterprise project type maintains a database on the server and uses it to generate a variety of live reports. You can see charts detailing the access frequency for each page, the number of unanswered questions posed to the system, the most common questions asked, and so forth. This gives you the opportunity to hone your help system to address demonstrated needs rather than the merely *anticipated* needs that all help systems aim to meet.

RoboHelp Enterprise can be overkill for developers who don't

## RoboHelp Enterprise

### eHelp Corp.

**Web:** [www.ehelp.com](http://www.ehelp.com) **Phone:** 800-358-9370

**Price:** \$1,798

**Quick Facts:** An all-encompassing help development system.

**Pros:** Powerful; covers a wide array of target platforms.

**Cons:** Natural language engine requires IIS; can't be deployed to standalone apps.





## A Web Site Brown Book

Microsoft products are complex and powerful, but they can be overwhelming. The only way to learn about them is to read the documentation. But the documentation is often too long and too technical. It's not always clear what you need to know, and it's not always clear where to find it. That's why we've created the Microsoft Web Site Brown Book. It's a simple, easy-to-use guide that tells you exactly what you need to know, and where to find it. It's the only book you need to get the most out of Microsoft products.

All I found in the book was a simple, easy-to-use guide that tells you exactly what you need to know, and where to find it. It's the only book you need to get the most out of Microsoft products. The book is written in a simple, easy-to-understand language that anyone can read. It's not full of technical jargon or complex diagrams. It's just a simple, easy-to-use guide that tells you exactly what you need to know, and where to find it. It's the only book you need to get the most out of Microsoft products.

For more information, visit our website at [www.microsoft.com](http://www.microsoft.com). We'll be there to help you every step of the way.

## Help is on the Way

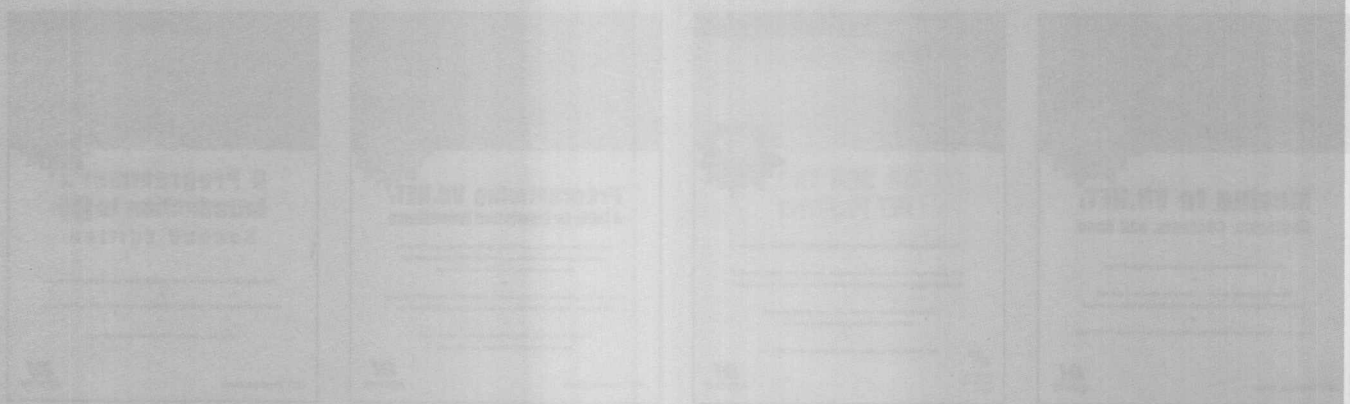
When you're having trouble with a Microsoft product, you don't want to wait for help. You want it now. That's why we've created the Microsoft Web Site Brown Book. It's a simple, easy-to-use guide that tells you exactly what you need to know, and where to find it. It's the only book you need to get the most out of Microsoft products.

The book is written in a simple, easy-to-understand language that anyone can read. It's not full of technical jargon or complex diagrams. It's just a simple, easy-to-use guide that tells you exactly what you need to know, and where to find it. It's the only book you need to get the most out of Microsoft products.

For more information, visit our website at [www.microsoft.com](http://www.microsoft.com). We'll be there to help you every step of the way.

**saved us billions of dollars!" often appear here.**

# The Ultimate .NET Resource



These books are the only ones you need to get the most out of Microsoft products. They are written in a simple, easy-to-understand language that anyone can read. They are not full of technical jargon or complex diagrams. They are just simple, easy-to-use guides that tell you exactly what you need to know, and where to find it. They are the only books you need to get the most out of Microsoft products.

## A Web Site Dream Team

Macromedia products are complex and state-of-the-art, which usually means the learning curve is steep. This suite is no exception, but the company is working to make its various products easier to learn and use. The suite's comprehensive documentation, instructional videos, help files, and extensive Web site do help navigate the learning curve, but don't expect to crank out a professional Web site after your first all-nighter.

All I found missing in this bundle are the features that Dreamweaver UltraDev—Dreamweaver 4's big brother—provides, such as easy data access and extensive support for programming. So big-time Web developers will probably want to spend the extra \$300 and get the UltraDev 4 Fireworks 4 Studio bundle. But if you're a designer who doesn't write code, this bundle is probably more than enough.

For the first time with a Web site development tool, I feel like I did when I first used Microsoft Office—it has many more features

than I'll ever use, but it's nice to have them available as I discover what they do. Sigh. It might finally be time to give FrontPage and Visual InterDev their pink slips. **VSM**

**Don Kiely** is a senior software technologist for Third Sector Technologies in Fairbanks, Alaska. He's written several books about VB and VC++, and he speaks regularly at VBITS. Reach him at donkiely@computer.org.

## Help is on the Way

need to target enterprise systems but do need to develop intensive help systems for standalone applications. Although RoboHelp Enterprise compiles help systems for HTML Help, WinHelp, JavaHelp, Oracle Help, and eHelp's browser-oriented platform-independent WebHelp format, you don't need it to target those formats. The RoboHelp Office version, which costs roughly half as much as the Enterprise version, tackles those formats exactly the same way as the Enterprise version. And,

because the Office version is a superset of the HTML Help and Classic versions, you can save even more by purchasing one of those two systems if you don't need to work with WinHelp files and the rest of the supported formats.

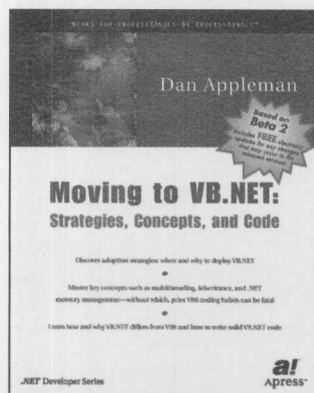
There's no way to address the fantastic array of development functions that all of these packages provide. In addition to creating help systems, you can also import and translate existing systems and output help systems to written manual format to take care of your printed documentation needs.

You can download 15-day evaluation copies from eHelp's Web site, so you should try the systems before committing to a purchase. Unless you work in an environment where no documentation is ever required, I think you'll enjoy working with RoboHelp. **VSM**

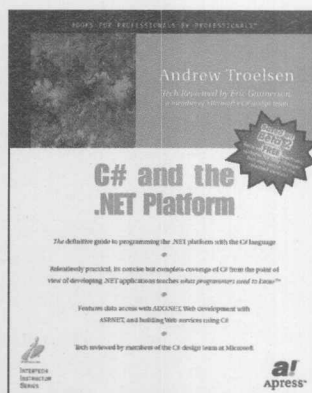
**Ron Schwarz** lurks in rural Michigan. When he's not digging out of the mountain of e-mail in his inbox, he maintains his sanity by restoring classic cameras. He welcomes your comments at [www.clubvb.com](http://www.clubvb.com).

# Apress<sup>TM</sup>

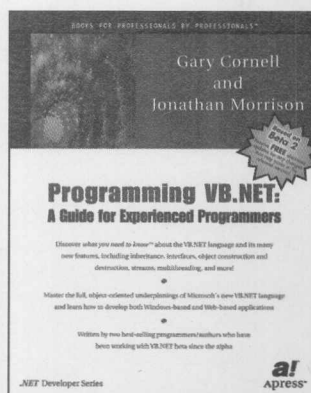
## The Ultimate .NET Resource



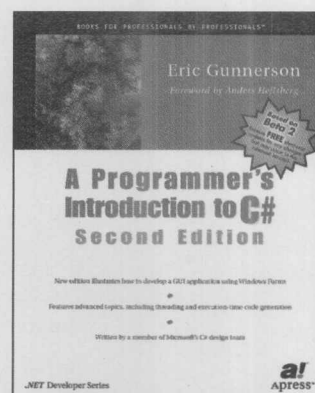
ISBN: 1-893115-97-6 \$39.95



ISBN: 1-893115-59-3 \$59.95



ISBN: 1-893115-99-2 \$39.95



ISBN: 1-893115-62-3 \$39.95

Apress, located in Berkeley, CA, is an innovative publishing company devoted to meeting the needs of existing and potential programming professionals. Simply put, the "A" in Apress stands for the "author's press<sup>TM</sup>." Apress' unique author-centric approach to publishing grew from conversations between Dan Appleman and Gary Cornell, authors of best-selling, highly regarded computer books. In 1998, they set out to create a publishing company that emphasized quality above all else — a company with books that would be considered the best in their market. Dan and Gary's vision has resulted in over 35 titles by many leading software professionals.

**Books for professionals by professionals<sup>TM</sup> • [www.apress.com](http://www.apress.com)**

Available at bookstores nationwide or from Springer-Verlag New York, Inc. at 1-800-777-4643; fax 1-212-533-3503.  
Contact us for more information at [sales@apress.com](mailto:sales@apress.com).

**a!**<sup>TM</sup>  
**Apress<sup>TM</sup>**

# Deliver Spreadsheets Over the Web

**E**xcelWriter 2.0 is a server-side component for Internet Information Server (IIS) that generates Microsoft Excel spreadsheets from a data source. Using Active Server Pages (ASP), you can produce text, graphs, and charts that correspond to the BIFF 8 format used by Excel 97, 2000, and 2002. This capability lets you take advantage of Excel's rich formatting and formulas while providing the latest statistics from your database. The budget-conscious should take note that you don't even need to buy Excel to run or view reports—the component works without requiring Excel on the Web server, and Microsoft offers a free Excel reader.

During the easy installation, watch for a privacy issue. The

## ExcelWriter 2.0

### SoftArtisans

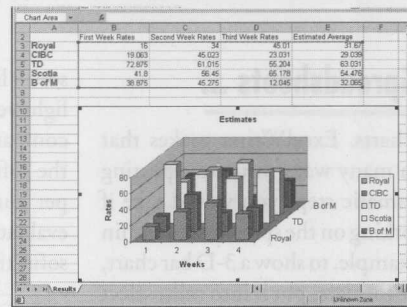
**Web:** [www.softartisans.com](http://www.softartisans.com) **Phone:** 877-SOFTART

**Price:** Server license starts at \$799.

**Quick Facts:** Component that generates Excel spreadsheets, reports, and charts from scripts or databases.

**Pros:** Flexible and sophisticated; built for high-powered Web use; good documentation and sample code.

**Cons:** No support for pivot tables; sends registration information to SoftArtisans without prompting; no fixed price.



**Generate Excel Spreadsheets—Without Excel.** ExcelWriter 2.0 lets your Web server deliver Excel spreadsheets on the fly without requiring Microsoft Excel on the server. You can connect to an ADO data source and program ExcelWriter's objects using ASP to generate 2-D and 3-D charts.

license says the product sends your registration data to SoftArtisans automatically. Specifically, "HRESULT of registration, Username, Computername, Domainname, Time Zone, OS Version, Locale Id, Browser Version, Source IP Address and CPU count."

To give ExcelWriter a spin, I used ASP to read data from an Access database and push the results out to Excel as a sorted table. Thanks to the example files, producing a simple spreadsheet application takes only a few minutes. Use the familiar ActiveX Data Objects (ADO) to connect to the database and execute a SQL query. After assigning values to the cells, you define the style of the columns using ExcelWriter's CreateStyle method. For example, you can format a date using DateStyle.Number = 17, or you can set the width of a text column using cells.ColumnWidth(i) = 20.

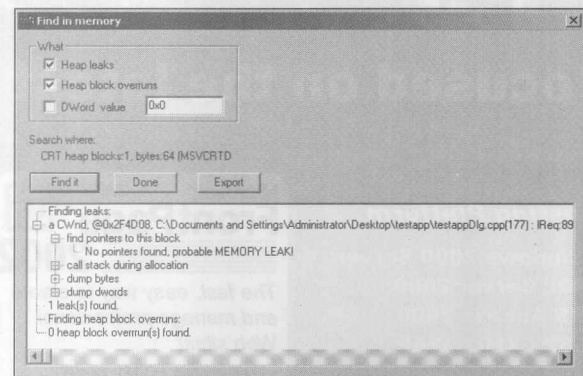
If you're generating Excel spreadsheets, you probably also want

# Snuff Out Resource Leaks

**F**inding bottlenecks and resource leaks in your code is a crucial part of testing and deploying robust applications. There are many choices out there for tools that can help you with these tasks. Most of them are large, complex, and costly.

But Electric Software's GlowCode is an excellent tool for profiling and understanding your code and detecting memory leaks at an affordable price. The tool is simple and straightforward to use, and I was able to get meaningful, detailed, and easily understandable results in minutes the first time I used it.

Using GlowCode starts with the GlowCode Loader, which allows you to choose whether you want to launch your app from the loader directly, attach the GlowCode Profiler to a running process,



**GlowCode Finds a Leak.** GlowCode detects leaked resources quickly and easily, providing detailed information to help track down the cause.

or link the GlowCode toolset directly into your application. If you choose the latter, you get some brief instructions to add the provided glowcode.cpp file to your app. But you also need to ensure the GlowCode gc2core.dll file is either in your program's folder or in a system path folder where you can find it. If you're going to use GlowCode with a release version of your program, you also need to make sure a program database (PDB) is available.

Once GlowCode is attached to a running application through any of these means, the GlowCode Dashboard arrives to provide information on the running program. The Dashboard is an easy-to-use tabbed interface with tabs for each of the program's main functions: Profiler, Trace, Coverage, and Resources. There's also a More tab, which is where you find the basic program functions

## GlowCode 3.0

**Electric Software Inc.**

**Web:** [www.glowcode.com](http://www.glowcode.com)

**Price:** \$295

**Quick Facts:** Quick and easy profiling tool that snuffs out resource leaks and bottlenecks.

**Pros:** Fast, easy to learn and use.

**Cons:** Slightly primitive user interface; sparse online documentation.



### Deliver Spreadsheets ...

to display charts. ExcelWriter makes that easy, too. In many ways, you are repeating the programmatic steps you would take if you were working on the spreadsheet within Excel. For example, to show a 3-D bar chart, you generate the data, put it into cells, select ranges, and set the format for the type of chart. Anyone who is comfortable in VBA will grasp the objects in ExcelWriter easily. One disappointment is its lack of support for pivot tables. That addition would give it a big competitive boost against reporting products from Crystal Decisions.

In my examples, I generated spreadsheets on the fly. For static reports, schedule a Windows Scripting Host script to write a single XLS file to the file system instead of streaming new spreadsheets to each user.

ExcelWriter is fun to use, works well, and provides good documentation and example code. For the forward-thinking purchaser, SoftArtisans' Web site shows the product working already in ASP .NET. A

server license starts at \$799 per CPU for the light version, but you'll have to contact the company on pricing out the full version of the software because it is "negotiated on a per purchase basis." You can obtain a free evaluation download from <http://support.softartisans.com/evalindex.asp>. **VSM**

**Ken Cox** is a technical writer and Web applications developer in Toronto. A former broadcast journalist, Ken is also a Microsoft MVP. Reach Ken at [kjopc@hotmail.com](mailto:kjopc@hotmail.com).

### Snuff Out Resource Leaks

normally found in a menu, such as Help, options, version info, and so on.

The Profiler, Trace, and Coverage tabs take a consistent approach, consisting of an expandable treeview display that lets you drill down to the module, file, function, or line of code level. There are also additional columns in the tab that display the results of that tab's purpose. Each tab has buttons to

start and stop the profiling, clear the display and counts, change the current hooks into the program, export the displayed results to a file or the clipboard, or get options relevant to the tab's function. You can hook into system or linked DLLs that your program uses if you want to see their contribution to your code's performance.

You can use GlowCode as a simple tool during ad hoc testing to ensure you get good coverage of your code, or as a cross-check of test suites to make sure you're getting the coverage you think you are. The documentation is a little sparse and the user interface a bit primitive, but because the tool is so straightforward and intuitive, you don't need a fancy GUI or reams of documentation to get useful results in a hurry. **VSM**

**Brian Noyes** is a senior software engineer with Digital Access Corp. He's an MCSD with nine years of software development, project management, and test and evaluation experience. E-mail him at [bnoyes@domeworks.com](mailto:bnoyes@domeworks.com).

## Focused on the latest Microsoft® Technologies

Cluster Solutions • Dedicated Solutions • Shared Web Solutions • Application Solutions

### Advanced Hosting Platform

- Windows 2000 Server
- SQL Server 2000
- MS Access 2000
- ASP / XML / WAP
- ASP Components
- Live Statistics
- Active Directory
- HostMechanic™

### Cutting-edge Development Tools

- MS FrontPage 2002
- Visual Studio.NET

### Microsoft® FrontPage® 2002

The fast, easy way to create and manage professional Web sites!

- Generates HTML code.
- Easily add E-commerce functionality to your site.
- Automatically updated Web content (MSNBC, Expedia)
- 67 Customizable Themes updated for 2002!

### Microsoft® Exchange 2000 Server

Ideal platform for business critical messaging. Powerful collaborative features!

- Keep your company's staff up-to-date in real time!
- Unified management of all messaging, collaboration and network resources.
- Greater efficiency and power by unifying information flow.

### Microsoft® SharePoint™ Team Services

A team Web site solution designed to improve the way a team manages information!

- Seamless communication and efficient collaboration.
- Document Libraries.
- Discussion Boards.
- Customize from a Browser.
- Customize with Microsoft® FrontPage® Version 2002.

### Microsoft® .net

The Next Generation Internet...

Our research and development technicians have been working with .NET since its inception in preparation of supporting this new and innovative Microsoft® platform!



Toll Free: 800.621.8309  
[www.innerhost.com](http://www.innerhost.com)

Complex Hosting in a Global Environment™



## Product Listings

**Editor's Note:** Please send product information to Product Listings Editor, c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax: 650-853-0230; e-mail: [vsmedit@fawcette.com](mailto:vsmedit@fawcette.com).

### Dataphor

Dataphor is designed to provide responsive and thorough data access and manipulation. Written in C#, the program consists of a language specification and a server architecture. Features include unified navigational access, an extensible data dictionary, database-level constraints, and the ability to update queries. Contact for pricing.

#### Softwise

Phone: 888-324-3988; 801-221-8222

Web: [www.softwisetools.com](http://www.softwisetools.com)

### Gen<X>

Gen<X> is designed to capture code written into templates and reuse it from project to project. The program combines changing input variables with static code, using an engine to merge at run time, and a syntax to declare the variables when making a template. The syntax is virtually an exact match to Active Server Pages (ASP) and does not require a Web server to run. Contact for pricing.

#### DevelopMentor

Phone: 310-214-7800

Web: [www.develop.com](http://www.develop.com)

### Interphor

Interphor is a group of technologies that generate user interfaces dynamically using information provided by the data dictionary. The interfaces are then presented using a thin client metaphor. You can tailor complete applications to fit the requirements of any project. Contact for pricing.

#### Softwise

Phone: 888-324-3988; 801-221-8222

Web: [www.softwisetools.com](http://www.softwisetools.com)

## Updates:

### Acceler8DB 4.7

Acceler8DB is a client/server database management system designed for Windows and AS/400 systems. The program allows applications to access multiple Acceler8DB and DB2/400 databases concurrently and interchangeably. The program stores and presents data through physical and logical files. You can also store data on standalone PCs or on Windows NT/2000 servers accessible by TCP/IP. Acceler8DB is equipped with an Acceler8DB Manager to facilitate the administration of database and print files. Contact for pricing.

#### ASNA

Phone: 800-289-2762

Web: [www.asna.com](http://www.asna.com)

### AppForge 2.0

AppForge 2.0 enables Visual Basic developers to create applications for Palm OS, Pocket PC, and Win32-powered devices using a single code base. It also allows PDB libraries to run on the Pocket PC, giving cross-platform functionality. New features in version 2.0 include

color support, dropdown menus, user-defined classes, double-precision floating-point operation, overlapping control support, larger code segments, and currency support for Visual Basic 6.0. Contact for pricing.

#### AppForge

Phone: 678-686-9000

Web: [www.appforge.com](http://www.appforge.com)

### AppForge SDK 2.0

AppForge SDK 2.0 is a collection of files, libraries, and utilities that allow you to use Visual C++ to create applications and ActiveX controls for Palm OS, Pocket PC, and Win32-powered devices using a single code base. The ActiveX controls created with the SDK are designed to integrate seamlessly with AppForge 2.0. The program integrates with Visual Studio for Windows, Metrowerks CodeWarrior, or PRCTools for Palm OS. Contact for pricing.

#### AppForge

Phone: 678-686-9000

Web: [www.appforge.com](http://www.appforge.com)

### CAST Application Mining Suite 4.0

CAST Application Mining Suite 4.0 is a suite of tools that generates a graphical roadmap of client/server applications and databases, showing the structure of a system and the interrelationship of its objects. Its drill-down views and change-impact analysis are designed to help you gain control of large, complex database applications. It supports SQL Server and Oracle databases as well as applications written in Visual Basic and PowerBuilder. It also offers limited support for 3GL code stored as ASCII text, such as Visual C++. New features in version 4.0 include a new Java Analyzer, support for J2EE- and .NET-compliant applications, optimized legibility of graphical views, and a new HTML Report wizard. Contact for pricing.

#### CAST

Phone: 415-296-1300

Web: [www.castsoftware.com](http://www.castsoftware.com)

### Chart FX Client Server

Chart FX Client Server is a COM-based, 32-bit charting tool that lets you deploy powerful charts in your VB, Visual C++, Delphi, and Access applications. The program includes an OLE server, server extensions, and Client Server wizards. It is recoded completely in C# to ensure .NET compatibility. New features include increased security, integration with Web Services, enhanced XML support, new API and Flash generation, use of new GDI features, and adoption of the Windows XP OS look and feel. \$499.

#### Software FX

Phone: 800-392-4278; 561-999-8888

Web: [www.softwarefx.com](http://www.softwarefx.com)

### Chart FX IE 2000

Chart FX IE 2000 is a charting tool for Internet/intranet developers. It offers the same charting engine as Chart FX Client Server; an image painter; the ability to generate ASP, ColdFusion, and client-side code; and Visual Interface 6.0 compliance. New features include increased security, integration with

Web Services, enhanced XML support, new API and Flash generation, use of new GDI features, and adoption of the Windows XP OS look and feel. Starts at \$1,399.

#### Software FX

Phone: 800-392-4278; 561-999-8888

Web: [www.softwarefx.com](http://www.softwarefx.com)

### Database Toolbox

ASNA Database Toolbox connects ASNA's Acceler8DB database to Windows and Web applications. The toolbox is a scalable database management system designed specifically for Windows and Active Server Pages (ASP) applications, providing direct database access for the AS/400, Windows NT/2000 Server, and local PCs. You can use it with Visual Basic, Visual C++, and Java. Contact for pricing.

#### ASNA

Phone: 800-289-2762

Web: [www.asna.com](http://www.asna.com)

### DB2 Universal Database 7.2

DB2 Universal Database 7.2 is a Web-enabled relational database management system. It is scalable from single processors to symmetric multiprocessors to massively parallel clusters. New features in version 7.2 include new assist wizards, Sybase and SQL Server wrappers, a DB2 Connect Web Starter kit, and support for messages, including XML documents. Contact for pricing.

#### IBM

Phone: 800-IBM-4YOU; 914-499-1900

Web: [www.software.ibm.com](http://www.software.ibm.com)

### NT Service Toolkit 1.1

NT Service Toolkit 1.1 is a framework that lets you write fully featured NT/2000 services while adhering to Visual Basic's programming and threading rules. The program supports all NT service options and controls. It supports client requests on independent threads for scalability, with available client impersonation allowing services to act on behalf of clients in their own security context. Client requests and service control is possible through COM/COM+/DCOM. You can build monitors, background taskers, agents, resource pools, and business objects using Visual C++. \$499.

#### Desaware

Phone: 408-377-4770

Web: [www.desaware.com](http://www.desaware.com)

### R:BASE 2000 6.5

R:BASE 2000 6.5 is a PC-based database management system based on the relational model. It provides intuitive interfaces, such as clickable images, and user-defined functions that offer direct control of function specification. The engine allows for large, customizable data queries; you can use it over a network without setting up a network server file. Features in version 6.5 include support for long filenames, support for calling DLLs, form flexibility, the ability to use PAUSE 3 functionality and WALKMENUMs in a Windows environment, and context-sensitive help. Starts at \$500.

#### R:BASE Technologies

Phone: 724-733-0053

Web: [www.rbase.com](http://www.rbase.com)

### Rational Rose 2001a

Rational Rose 2001a is a graphical component modeling and development tool that facilitates software application modeling. The program lets you visualize your applications graphically using industry-standard UML. It supports generation and round-trip engineering in C++, Visual Basic, and Java. New features in version 2001a include the ability to forward- or reverse-engineer directly to an IDE, the ability to search for JDKs automatically, 20 new design patterns, enhanced J2EE support, support for modeling of views and tablespaces, and data storage modeling for DB2, Oracle, SQL Server, and SQL Server 2000. Contact for pricing.

#### Rational Software

Phone: 800-728-1212; 408-863-9900

Web: [www.rational.com](http://www.rational.com)

### ScheduleX 8.0

ScheduleX 8.0 combines five lightweight (ATL) components for calendar, task management, and scheduling applications. Advanced Calendar, DayView, and TaskPad components let you create Personal Information Management (PIM) software, project management and tracking, or any application that requires advanced scheduling. New features in version 8.0 include a configurable display, support for international date formats, customization of date separators, and the ability to choose the time display style. \$395.

#### Infragistics

Phone: 800-231-8588

Web: [www.infragistics.com](http://www.infragistics.com)

### Team Remote Debugger 2001

Team Remote Debugger 2001 lets you trace any number of code units of any kind (ASP, MTS, T-SQL, COM+, ActiveX EXE, DLL, COM, Thread, or CFML) written in any language (ASP, VB, VC++, Delphi, T-SQL, VJ, or CFML) residing on multiple shared and dedicated servers at the same time, without the need to attach to process. Features include the ability to pass messages from remote code units to local machines, the ability to pass dialogs from remote code, remote code debugging, debugging on multiple servers, and collaborative team debugging. Contact for pricing.

#### Spline Technologies

Web: [www.remotedebugger.com](http://www.remotedebugger.com)

### VersionStamper 6.5

VersionStamper 6.5 is an ActiveX control that detects component version incompatibilities on end-user systems. It lets you create self-updating apps with support for the Internet and intranet. The program includes both 16-bit and 32-bit COM/ActiveX OCX controls, and it works with all COM/ActiveX control-compatible development environments. New features in version 6.5 include the ability to include custom messages for file conflicts in a file list, support to run the Conflict Wizard in batch mode to generate or update the reference file list, support for scanning Wise and InstallShield installation scripts to generate the reference file list, and updates to the VersionStamper Internet components to support proxy servers. \$249.

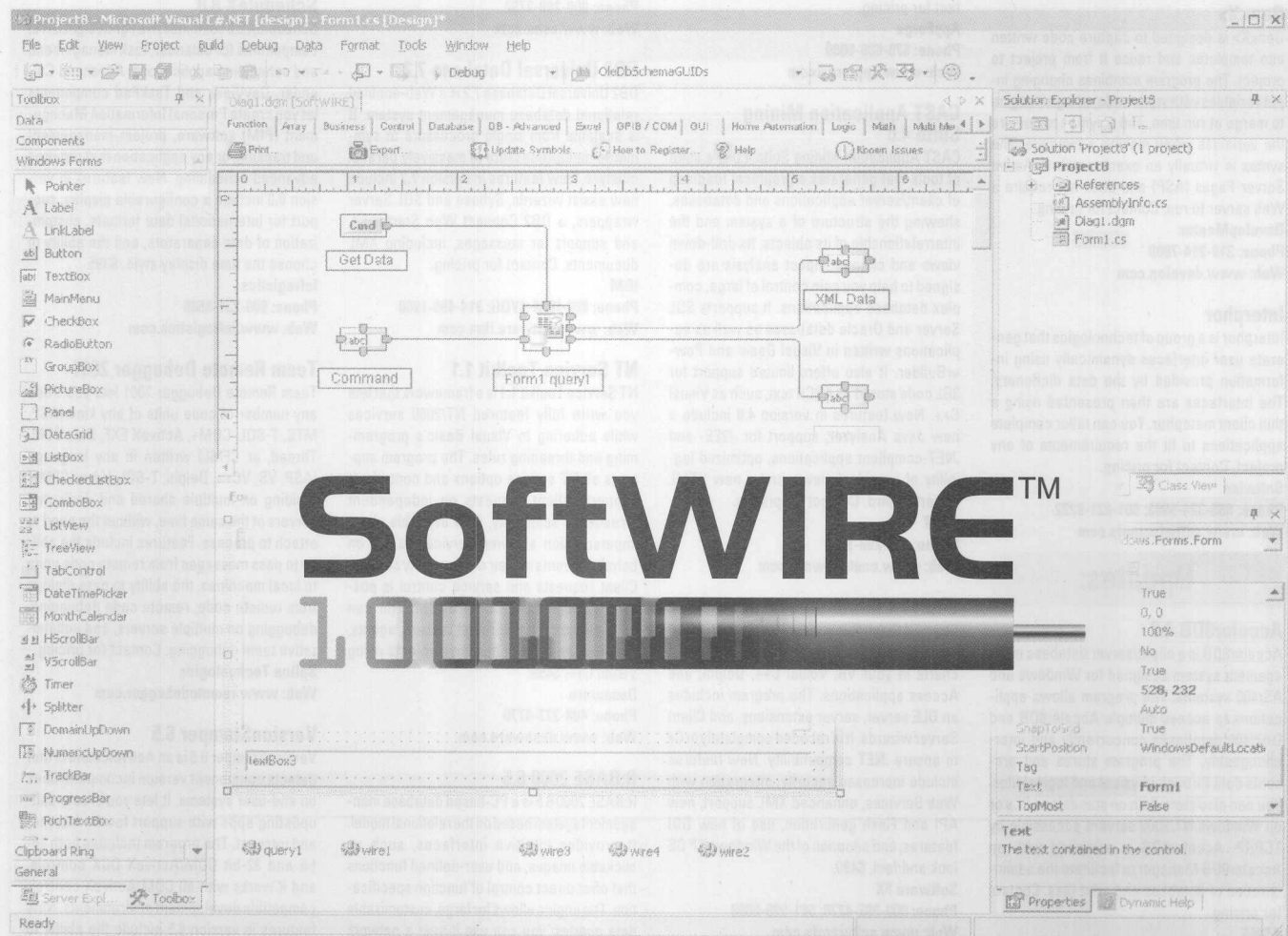
#### Desaware

Phone: 408-377-4770

Web: [www.desaware.com](http://www.desaware.com)

# add SoftWIRE.NET... your quickest route to Managed Code.

## Graphical programming for Visual Studio.NET



**Download SoftWIRE.NET now!**

**[www.softwire.com](http://www.softwire.com)**

©2001 by SoftWIRE Technology, Inc. All rights reserved.  
SoftWIRE and the SoftWIRE logo are trademarks of SoftWIRE Technology, Inc.  
NET is a trademark of Microsoft, Inc. All other trademarks are the property of their respective owners.

**SoftWIRE Technology, Inc**

16 Commerce Blvd • Middleboro, MA 02346  
phone: (508) 946-8900 • fax: (508) 946-9500

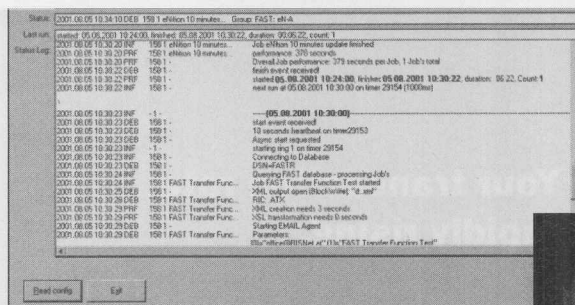


# Get Financial News FAST

The Organization of the Petroleum Exporting Countries (OPEC) and Austrian national newspapers are quick to gather their financial and world news using a Visual Studio solution called FAST (Financial Access Storage and Transfer system).

FAST is a back-office application that acquires data from the well-known Reuters news service—either by satellite or land line—and stores it in a database. It then delivers the data to the user either in raw XML or in formatted Extensible Stylesheet Language (XSL) through HTTP, FTP, Simple Mail Transfer Protocol (SMTP), or Short Message Service (SMS). FAST provides an “easy way to work with Reuters financial data,” says sole developer Richard Prinz of Regionale Informations Systeme GmbH (RIS) in Wiener Neustadt, Austria. Users can run the app in the background as a service or in the foreground as a console.

RIS uses FAST internally, delivering the output to subscribing customers. But the company will soon sell it to users who want to process Reuters data themselves. The 20 MB app, consisting of 20 DLLs and six EXEs, took Prinz one and a half years to develop. He



For more information on FAST, e-mail Richard Prinz at [richard.prinz@risnet.at](mailto:richard.prinz@risnet.at).



used Visual Basic 6.0 for 80 percent of the solution, and Visual C++ and Visual J++ for the rest. FAST supports Microsoft Access and SQL Server databases, and will support Oracle in the future.

Prinz used Reuters' Source/Sink Lib (SSL) development toolkit to build FAST, as well as “lots of information from MSDN and [the Microsoft] Knowledge Base,” he says. He's most proud of the app's flexibility and was most challenged by its “horrible” multi-threading issues.

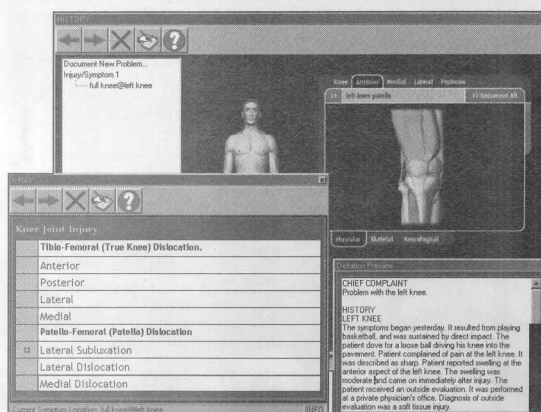
In the future, Prinz would like to add features for statistical data analysis and mobile usage. He's also actively evaluating Visual Studio .NET: “We have already started some development with the beta version and are satisfied with the results ... in about a year, if all goes well, we'll have a .NET version available.” —Nina Goldschlager

# Handhelds Visit the Doctor's Office

Handheld computers seem to be popping up everywhere now—including the doctor's office. Typically, doctors meeting with patients dictate to a recorder or scribble notes on a pad. After the visit, someone transcribes the notes, then a medical coder reviews them and applies appropriate codes for billing. This system can be time-consuming, costly, and error-prone.

PICOS, based in Sterling, Va., has circumvented this process with the Dictation Engine—part of PICOS' Total Practice Manager (TPM) suite, which performs all the tasks needed at a doctor's office, including scheduling, billing, purchasing, and inventory. Physicians at Dulles Orthopedics Group—a beta test site in Sterling—employ TPM on their Fujitsu Stylistic LT C-500s. As doctors meet with patients, they point-and-click through a series of logically displayed questions and answers, many displayed as detailed, high-resolution, 3-D graphics of the human anatomy. Once doctors finish their exam, the Dictation Engine produces a readable dictation that becomes part of the patient's history, as well as codes that meet industry standards for billing. Only a simple review and approval from billing personnel are needed to send the claim to the patient's insurance company.

Stephen Mallette, PICOS' chief information officer, designed and developed the Dictation Engine in VB6 over a period of five months. System developer James Policano implemented the app in its host application, TPMVisit, a module of TPM. Satish Unni,



For more information on the Dictation Engine, visit PICOS at [www.picosinc.com](http://www.picosinc.com), or reach Stephen Mallette by e-mail at [smallette@picosinc.com](mailto:smallette@picosinc.com).



system analyst, gathered and documented the logic that drives the engine. The Dictation Engine comprises 10 class modules and 15 forms (roughly 750 possible forms if you count those generated dynamically from the logic files).

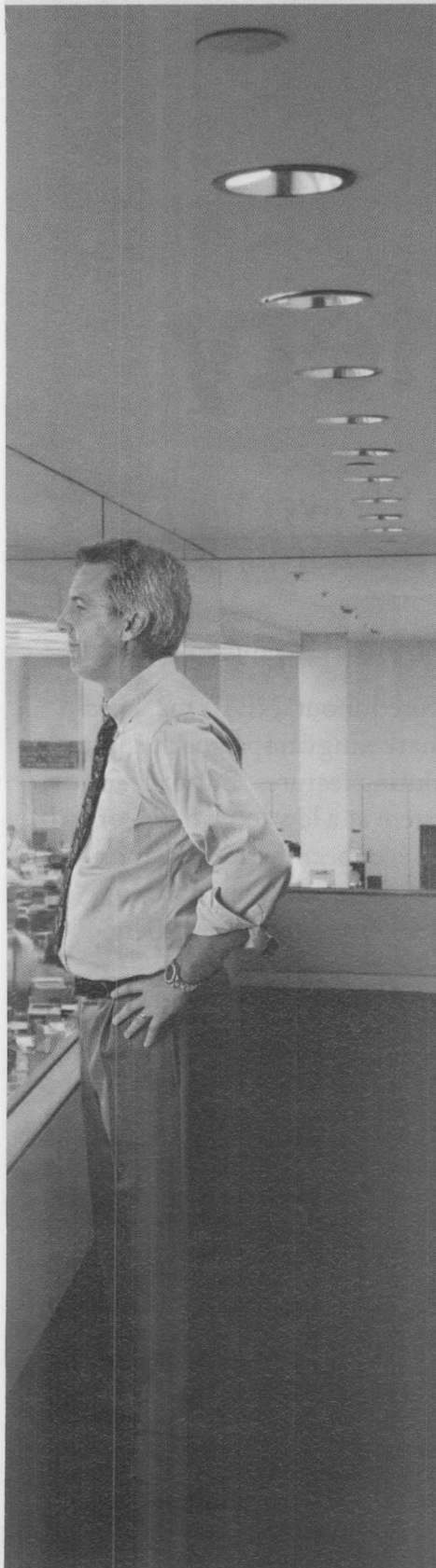
Mallette is proud that “the project came in on time, under budget, and with all requested functionality present.” He adds: “The key to our success was not underestimating the level of effort required.” Future plans include adding logic files for other medical specialties besides orthopedics. Another possibility: developing a logic file for capturing sporting-event statistics. —Nicole Pizzurro

To submit proposals for VS All-Stars, send an e-mail to [vsmedit@fawcette.com](mailto:vsmedit@fawcette.com).

**Your anxiety rate:  
steadily falling.**







**MICROSOFT® SQL SERVER™ 2000** The company continues to grow and grow. And your system's transaction rates are climbing in lockstep with it. A big deal, for sure—but nothing to get anxious about. That's because SQL Server 2000 offers scalability for even the most demanding environments. So you know that no matter how much—or how fast—your system needs to expand, you'll have the tools to handle it.

Part of the flexible Microsoft .NET Enterprise Server family, SQL Server 2000 recently achieved 20,000 SAP Standard Sales and Distribution (SD) Application Benchmark Users with a response

SQL SERVER 2000 BENCHMARKS

Application/Workload	World Record Results
SAP R/3 Sales & Distribution	20,000 concurrent users
TPC-C	688,220 tpmC <sup>††</sup>
PeopleSoft 8 CRM	30,000 concurrent users
PeopleSoft Financials	13,000 concurrent users
J.D. Edwards OneWorld	4,500 concurrent users
Onyx	32,000 concurrent users

time of 1.91 seconds on a 32-processor Unisys ES7000 system running as the database server

in a R/3 4.6C 3-Tier environment, surpassing a result achieved with Oracle 8.1 running on a Sun system with 64 processors.\* In addition to world-class performance, SQL Server 2000 has price/performance numbers that are 2.5 times better than the closest competitor.<sup>†</sup> Add to that native XML support in SQL Server 2000 and you've got the database that's ready to handle Web-enabled enterprise applications far into the future.

To get the full story on Microsoft SQL Server 2000 scalability, go to [microsoft.com/sql/worldrecord](http://microsoft.com/sql/worldrecord) **Software for the Agile Business.**

**Microsoft®**

Users: \*20,000 SAP SD Standard Application Benchmark certified by SAP 6/12/2001. Unisys ES7000: 1.91 second Average Response Time (Dialog), 6,046,000 Dialog Steps/Hour, 2,015,330 line items per hour, SAP R/3 4.6C, 32x 900MHz Pentium III Xeon AP 3/31/2000. Sun E10000: 1.88 second Average Response Time (Dialog), 5,864,000 Dialog Steps/Hour, 1,954,670 line items per hour, SAP R/3 4.0B 64x 400MHz UltraSparc II processor, 64GB memory, DBMS: Solaris 8, Oracle 8.1.5. <sup>†</sup>Best SQL tpmC, 22.58 \$/tpmC, available 5/31/01. TPC is a trademark of the Transaction Processing Performance Council, [www.tpc.org](http://www.tpc.org). TPC results as of 6/12/01.



# Pointers Add Power and Safety

VB.NET comes as close to having pointers as VB ever has, while making memory manipulation safer at the same time.

by Robert Teixeira

## Technology Toolbox

- ☒ VB.NET
- ☐ C#
- ☐ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☐ VB6
- ☒ Other:  
.NET SDK

## Resources

- .NET Framework Class Library: [http://msdn.microsoft.com/library/en-us/cpref/html/cpref\\_start.asp](http://msdn.microsoft.com/library/en-us/cpref/html/cpref_start.asp)
- "Break API Barriers" by Robert Teixeira [*Visual Basic Programmer's Journal* August 2001]
- Black Belt Programming, "Provide Pointers to Class Functions," by Matthew Curland [*Visual Basic Programmer's Journal* August 2001]

**M**any VB programmers are concerned and confused about .NET's managed memory model. Perhaps this is due partly to marketing campaigns that put the spotlight on .NET's Web and distributed computing features. But make no mistake: VB's ability to work with the Windows system at a low level, and work with unmanaged code from external libraries, has never been better. I demonstrated some of VB.NET's powerful new features in this area in my "Breaking API Barriers" article (see Resources), but that was only the tip of the iceberg.

In this article, I bid farewell to *CopyMemory* (also known as the *RtlMoveMemory* API function) and show you VB.NET's amazing new power over memory and pointers. That power is found in four new tools: *IntPtr*, .NET's platform-dependent representation of a memory address; *GCHandle*, which helps you pin and retrieve the address of data in the managed memory heap; *the Marshal class*, the one-stop shop for all your memory allocation, cleanup, and manipulation needs; and *Delegate*, .NET's representation of a function pointer.

It's worth going over a few safety issues before you dive into .NET's new pointer features. I covered one of those issues in my August article, but I'll restate it here because it's worth review within this context. An automatic garbage collector (GC) maintains the managed heap: It compacts unused space and readjusts any managed references to the memory that was moved. Because of its dynamic nature, you need to take extra precautions whenever you want to capture an address within the managed heap. This involves the act of "pinning," which I'll revisit later in this article.

The next safety issue relates to data integrity. When you take the address of a piece of data within the managed heap and pass that address to

unmanaged code (any code written outside of .NET), the .NET system can't guarantee the integrity of your data because the unmanaged code can do whatever it wants to the data without considering safety. Therefore, the compiler marks any operations that do this (using API function calls, for example) as unverifiable.

The last safety issue involves trust and permissions. The .NET system assures that its elements won't compromise your computer's security. For example, it knows which resources require file-system access and allows you to grant or deny the use of this resource depending on where the code came from. However, the .NET system has no way of knowing what resources will be accessed by unmanaged code, or how maliciously the unmanaged code might use a given resource. What's more, by manipulating pointers directly, it's possible that the data or functions of a component marked as safe could be inadvertently or intentionally replaced by more malicious code—one more reason that calls to unmanaged code and direct pointer manipulation are marked as unverifiable.

If you decide to work with direct memory manipulation in VB.NET, the first thing you need to understand is the *IntPtr* type. *IntPtr* is a structure that

represents both addresses and handles (most handles are pointers to Windows pointers). IntPtr instances are also platform-dependent (or independent, depending on your point of view). On a 32-bit system, IntPtr is 32-bit, while on a 64-bit system, IntPtr is 64-bit. The benefit is that you don't need to change or recompile your code for both platforms. Any function within the .NET Framework that exposes a way to work with addresses and handles uses the IntPtr type. Such functions are marshaled to unmanaged code simply as the internal address number, meaning you can pass a variable of IntPtr type ByVal to any unmanaged code that expects a pointer. So the good news is that, although you can't use this in VB.NET:

```
Dim lpMyVariable As Void *
```

this works brilliantly:

```
Dim lpMyVariable As IntPtr
```

Note that there was no IntPtr type in beta 1—instead, functions that worked with pointers and handles used the Integer type. The IntPtr type has a ToInt32 method that converts the address to an Integer, but that can and will cause an Overflow exception on a 64-bit system. IntPtr also has a ToInt64 method, but you'll have to keep track of the platform if you want to do these conversions.

## Get a Handle on the Garbage Collector

Next, let's examine how you place the address of a variable into an IntPtr. You use the GCHandle class, which has an AddrOfPinnedObject method that returns an IntPtr for a variable. You must "pin" the data before you can get this address. This prevents the garbage collector from moving the data inadvertently while you're referring to the original address. This code pins a variable, displays its address on the console window, and frees the handle:

```
' dim managed variable
Dim MyString As String
```

```
' pin variable and create
' GC handle instance
Dim gh As GCHandle = GCHandle.Alloc _
    (MyString, GCHandleType.Pinned)

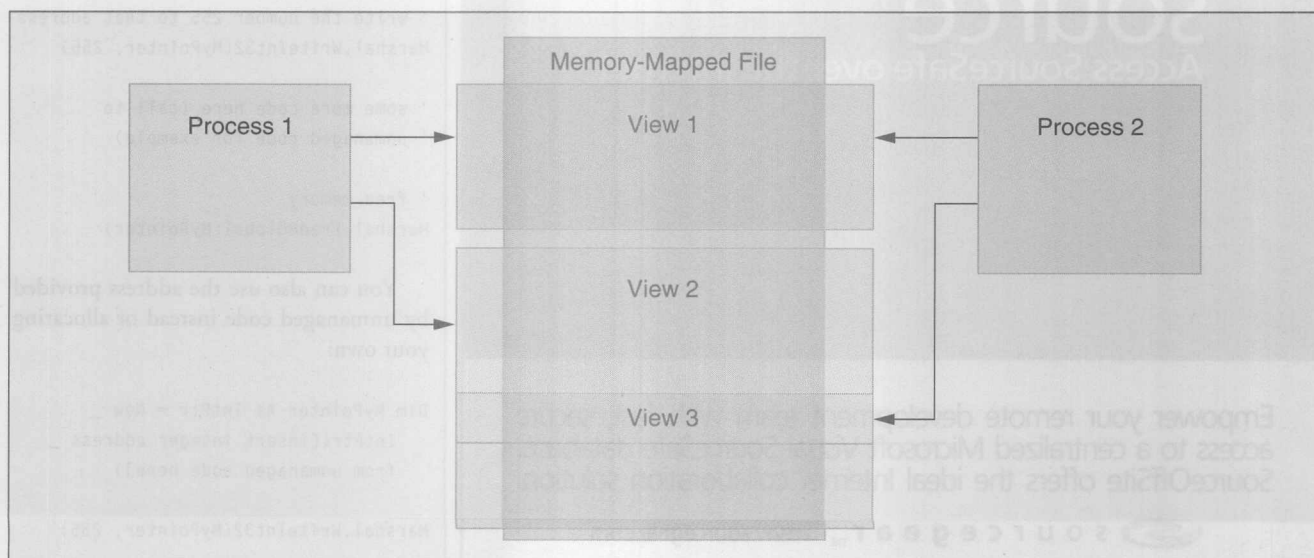
' get address of variable
Dim AddrOfMyString As IntPtr = _
    gh.AddrOfPinnedObject()
```

```
Console.WriteLine _
    (AddrOfMyString.ToString())
```

```
' free the handle and unpin variable
gh.Free()
```

Freeing a GCHandle as soon as you're done using it minimizes the impact on garbage collection. As a rule of thumb, this code shows the standard way to capture the memory address of a managed object. Note that you don't necessarily need to use this method to pass the address of a variable to an unmanaged function (such as a Windows API function). VB uses the .NET PInvoke service to pass parameters to unmanaged functions, and this feature has greatly enhanced VB's ability to pass data and addresses seamlessly to such functions. In this case, if you wanted to send the string's pointer to an external unmanaged function, you would be better off passing it in the parameter list as a String and using the MarshalAs attribute to specify the type of string. PInvoke takes care of the pinning automatically (see the "Break API Barriers" article for more details).

Be careful when you use value types, which are usually primitives or simple data types such as Integers. To utilize polymorphism in .NET fully, a value type needs to be able to work as object. VB.NET uses a technique called boxing to accomplish this. A value type is boxed when you implicitly or explicitly cast it as an Object in VB.NET. Therefore, value types are boxed when you use the Alloc method of the GCHandle class, because the Alloc method accepts an Object type parameter. If the address is



**Figure 1 Share Memories Between Processes.** Memory-mapped files are a powerful technique you can use to share "memory" between different processes. Processes can map to the same view or different views, or they can even have overlapping views.

being passed out, the unmanaged code still sees the correct value. However, if the unmanaged code writes back to the address, the new value won't carry over to the value type itself. This scenario is the same in C#, and its documentation states that if you need to write back to the value, you need to start off with an object instead. It looks like this in VB:

```
Dim MyVariable As Object = 123
```

The integer value type is cast to an object and boxed, and you can now pin the object representation of the integer. The value remains unchanged if you write back to this address. The object can then be cast back to an integer if needed. However, as I mentioned earlier, if you need to pass the address of a value type to unmanaged code, `PInvoke` is the most efficient way to do it—in other words, you pass the parameter `ByRef` to the unmanaged code.

### Memory Manager With Muscle

You now know how to use the `IntPtr` type and capture the address of a managed variable, but addresses aren't really useful unless you can *do* something with them. If you need the power of direct memory manipulation, then the `Marshal` class is your friend. This class has nothing but `Shared` (Static) functions, most of which serve as memory-management utilities. An added benefit of the `Marshal` class memory manipulation methods is that they're much safer to

use than arbitrary calls to something like `RtlMoveMemory`. Your need for this type of direct memory manipulation should be diminished in VB.NET, but later in the article I provide some examples of cases where you'd want to use it. The `Marshal` class includes several features worth mentioning. It enables you to allocate memory, as well as to read/write simple data, strings, user-defined types, and array data.

You can allocate memory on the unmanaged heap, fill this memory however you like, and pass its address to unmanaged code. The `Marshal` class has two methods to allocate memory: `AllocHGlobal`, which allocates memory on the native heap by using the `GlobalAlloc` function internally; and `AllocCoTaskMem`, which is similar but instead uses the COM memory manager (`CoTaskMemAlloc`). Both functions have one parameter, the number of bytes to allocate. Both functions return an `IntPtr`, the base address of the newly allocated buffer. To free the memory, you use the `FreeHGlobal` or `FreeCoTaskMem` methods, depending on which allocation method you used. Each of these functions has one parameter, the address to the newly allocated buffer returned by the allocation functions.

Use the `WriteByte`, `WriteInt16`, `WriteInt32`, or `WriteInt64` methods to write simple numeric data to an unmanaged buffer. Think of these as safer and more convenient versions of `CopyMemory` (`RtlMoveMemory`). Each of the functions takes as parameters the destination address of the write operation and the

numeric value you want to write. These functions are also overloaded to allow an optional third parameter indicating an offset from the address provided, which can be useful if you're attempting to fill memory with arrays of elements or fields of a structure, for instance:

```
' allocate some memory and get its address
Dim MyPointer As IntPtr = _
    Marshal.AllocHGlobal(4)
```

```
' write the number 255 to that address
Marshal.WriteInt32(MyPointer, 255)
```

```
' some more code here (call to
' unmanaged code for example)
```

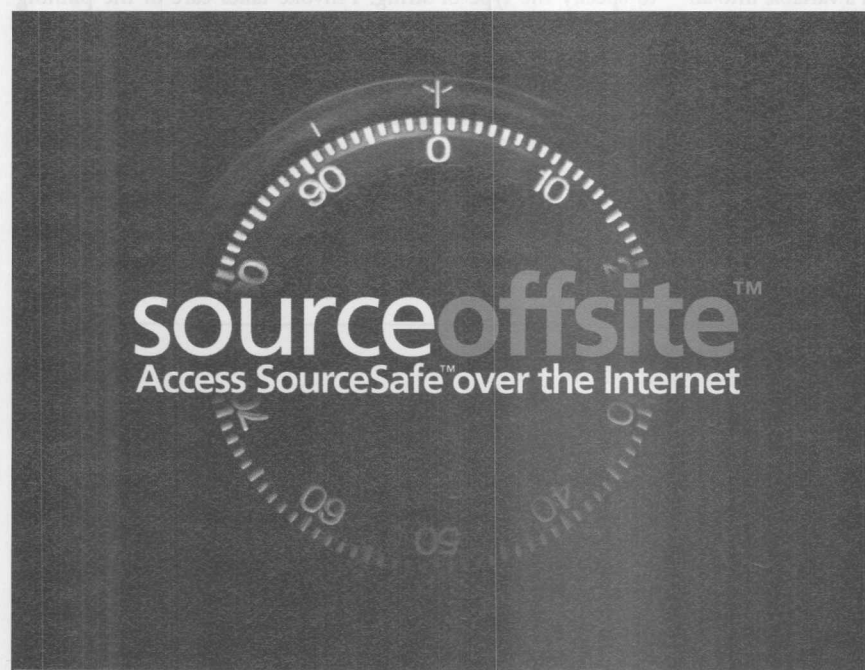
```
' free memory
Marshal.FreeHGlobal(MyPointer)
```

You can also use the address provided by unmanaged code instead of allocating your own:

```
Dim MyPointer As IntPtr = New _
    IntPtr([insert integer address _
    from unmanaged code here])
```

```
Marshal.WriteInt32(MyPointer, 255)
```

The reverse is also possible: You can read simple numeric data from an `IntPtr`



**sourceoffsite™**  
Access SourceSafe™ over the Internet

Empower your remote development team with fast, secure access to a centralized Microsoft Visual SourceSafe database. SourceOffSite offers the ideal Internet collaboration solution.

**sourcegear™** [www.sourcegear.com](http://www.sourcegear.com)

Visual SourceSafe is a registered trademark of Microsoft. Must be a licensed user of Microsoft Visual SourceSafe.



# ComponentOne Suites

## Sweet dreams.

High-performance suites for Visual Studio development.

Dreams really are sweeter with the ComponentOne family of Visual Studio® suites. That's because our feature-rich tools save time and money, which means less stress and less 12-hour workdays for you. Whether you choose a 2-product set or our jam-packed 14-product Enterprise Suite, you can rest easy knowing that we're dedicated to providing you with the most reliable and powerful tools on the market. Trust the performance, support, and value you can only get from ComponentOne tools. Download your eval copies today!



### ComponentOne Enterprise Suite

- True DBGrid® Pro 7.0
- VSFlexGrid® Pro 7.0
- VSVIEW® 7.0 Reporting Edition
- ComponentOne WebChart™ 7.0
- ComponentOne True DBList™ Pro 7.0
- ComponentOne SizerOne™ 7.0
- ComponentOne Query™ 1.0
- VSSPELL™ 6.0 International Edition
- True DBInput™ Pro 6.0
- True DataControl™ 6.0
- True DBReports™ 6.0
- True Help™ 1.0
- VSDOCX™ 1.0
- VSFORUM™ 2.0

### ComponentOne Developer's Suite

- True DBGrid® Pro 7.0
- VSFlexGrid® Pro 7.0
- VSVIEW® 7.0 Reporting Edition
- ComponentOne True DBList™ Pro 7.0
- ComponentOne Chart™ 7.0
- ComponentOne SizerOne™ 7.0
- True DBInput™ Pro 6.0
- True DataControl™ 6.0
- True DBReports™ 6.0

### ComponentOne Grid Suite 7.0

- True DBGrid® Pro 7.0
- VSFlexGrid® Pro 7.0

### True DBSuite™ 6.7

- True DBGrid® Pro 7.0
- ComponentOne True DBList™ Pro 7.0
- True DBInput™ Pro 6.0

### Combo Pak™ Plus 7.0

- VSFlexGrid® Pro 7.0
- VSVIEW® 7.0 Classic Edition
- ComponentOne SizerOne™ 7.0

### Combo Pak™ 7.0

- VSFlexGrid® Pro 7.0
- VSVIEW® 7.0 Classic Edition

TAKE ADVANTAGE OF OUR FABULOUS LIMITED TIME OFFERS:

ComponentOne Enterprise Suite.....	\$2,295. <sup>00</sup>	True DBSuite 6.7.....	\$799. <sup>95</sup>
ComponentOne Developer's Suite.....	\$1,795. <sup>00</sup>	Combo Pak Plus 7.0.....	\$649. <sup>95</sup>
ComponentOne Grid Suite 7.0.....	\$499. <sup>95</sup>	Combo Pak.....	\$499. <sup>95</sup>

## ComponentOne

www.componentone.com

East Coast Office: 1.800.858.2739 or 1.412.681.4343 • West Coast Office: 1.888.228.4839 or 1.510.496.3240

ComponentOne™ LLC • 4516 Henry Street • Pittsburgh, PA 15213 • USA

© 2001 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.

address using the `ReadByte`, `ReadInt16`, `ReadInt32`, and `ReadInt64` methods:

```
Dim MyInteger As Integer = Marshal.ReadInt32(MyPointer)
```

## Read/Write Strings

Reading and writing strings is similar to reading and writing simple numeric data, with one minor exception: You don't allocate memory first, then write a string to it. Instead, the act of creating a string in unmanaged memory allocates the space and returns the string's address. There are seven write methods: `StringToBSTR`, `StringToCoTaskMemAnsi`, `StringToCoTaskMemUni`, `StringToCoTaskMemAuto`, `StringToHGlobalAnsi`, `StringToHGlobalUni`, and `StringToHGlobalAuto`. The `StringToCoTaskMemxxx` functions write string data to COM allocated memory, while `StringToHGlobalxxx` functions write to the native unmanaged heap. Functions ending in `Ansi` write single-byte ANSI strings. Functions ending in `Uni` write double-byte Unicode strings. The functions ending in `Auto` write ANSI or Unicode strings, depending on the operating system: ANSI strings on Windows 98 and Me, Unicode strings on NT-based platforms (Windows NT 4.0, 2000, and XP). `StringToBSTR` writes an Automation BSTR, which is analogous to using the `SysAllocString` function. Each of these functions accepts a string as an input argument and returns a pointer to the resulting string:

```
Dim MyStrPointer As IntPtr = _
    Marshal.StringToHGlobalAuto("Hello World")
```

Four read methods—`PtrToStringAnsi`, `PtrToStringUni`, `PtrToStringAuto`, and `PtrToStringBSTR`—read the data at a given address and create a managed `String` object containing a copy of the characters. Use `PtrToStringAnsi` if the unmanaged string is ANSI, `PtrToStringUni` if the unmanaged string is Unicode, or `PtrToStringBSTR` if the unmanaged string is of BSTR type. `PtrToStringAuto` assumes the unmanaged string is ANSI on a Windows 98 or Me system, and Unicode on a Windows NT 4.0, 2000, or XP platform. `PtrToStringAuto` actually calls `PtrToStringAnsi` or `PtrToStringUni`, depending on the operating system. Each function is also overloaded to accept an optional number of characters to copy. If you don't provide the number of characters, the function looks for a terminating null character:

```
' copy entire string
Dim MyString As String = _
    Marshal.PtrToStringAuto(MyPointer)

' copy first 5 characters
Dim MyString As String = Marshal.PtrToStringAuto _
    (MyPointer, 5)
```

To free the unmanaged memory buffer holding a string, you call either the `FreeHGlobal` or `FreeCoTaskMem` member. To free a BSTR created with `StringToBSTR`, you call `FreeBSTR`, which in turn calls the `FreeSysString` function.

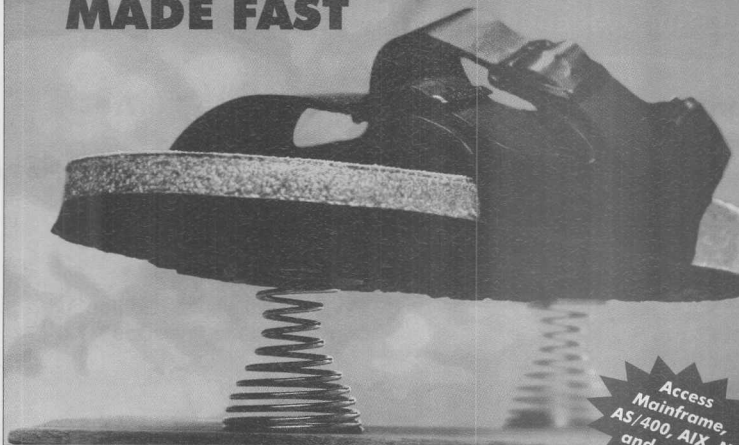
You write a Structure (user-defined type) to unmanaged memory

by using the `StructureToPtr` method. This method requires you to have a memory buffer allocated ahead of time. It takes three parameters: the Structure you wish to write, the address (`IntPtr`) of the memory buffer, and a delete flag. Setting the delete flag to `True` wipes and frees any existing data from the buffer. This is important, because you can cause memory leaks by failing to delete the existing buffer space. For example, if one of the fields of the structure is a reference to another structure or string, the data being referenced by the field won't be freed if the parameter is set to `False`. Also note that Structures are copied to unmanaged memory using specific formatting (which you can control optionally), and the unmanaged copy might not look exactly like the managed representation (refer to the "Break API Barriers" article for more information). Use the `SizeOf` method to determine the number of bytes required for the buffer:

```
Dim MyVariable As Point
MyVariable.X = 100
MyVariable.Y = 250

Dim MyPointer As IntPtr = _
    Marshal.AllocHGlobal _
    (Marshal.SizeOf(MyVariable))
Marshal.StructureToPtr(MyVariable, _
    MyPointer, False)
```

## DB2 DATA ACCESS MADE FAST



**Simply, Easily, for all DB2 Servers**

*HiT OLEDB/DB2 and HiT ODBC/DB2 middleware are fast, easy solutions to your performance needs. Direct DB2 connectivity. No gateways. Scalable. Latest ADO compatibility and SSL v3 for security. For fast XML data access, check out our Allora platform. Jump to it!*

**HIT SOFTWARE**  
Open Up Your Data

[www.hitsw.com](http://www.hitsw.com)

1-408-345-4001

[info@hitsw.com](mailto:info@hitsw.com)



Use the `PtrToStructure` method to reverse the process and read a structure from unmanaged memory. You can use this method either as a Function that returns a copy of the structure based on Type, or as a Sub that fills a structure parameter. Note that `PtrToStructure` returns an Object type reference, and with Option Strict On, you must cast to the structure type (by using CType, for example):

```
Dim MyPoint As Point
MyPoint = CType(Marshal.PtrToStructure _
    (MyPointer, GetType(Point)), Point)
```

### Read/Write Array Data

Reading and writing array data is especially valuable when you need to stream binary data. The `Copy` method does both reads and writes, depending on the parameters you pass to it. If you want to write the data, you need to allocate some buffer space first, just as you would with strings. Next, you call the `Copy` method and pass the array itself, the index in the array of the element you want to start the copy with, the destination address (IntPtr resulting from the allocation), and the size of the buffer:

```
Dim MyData(255) As Byte
' insert code here to fill byte array

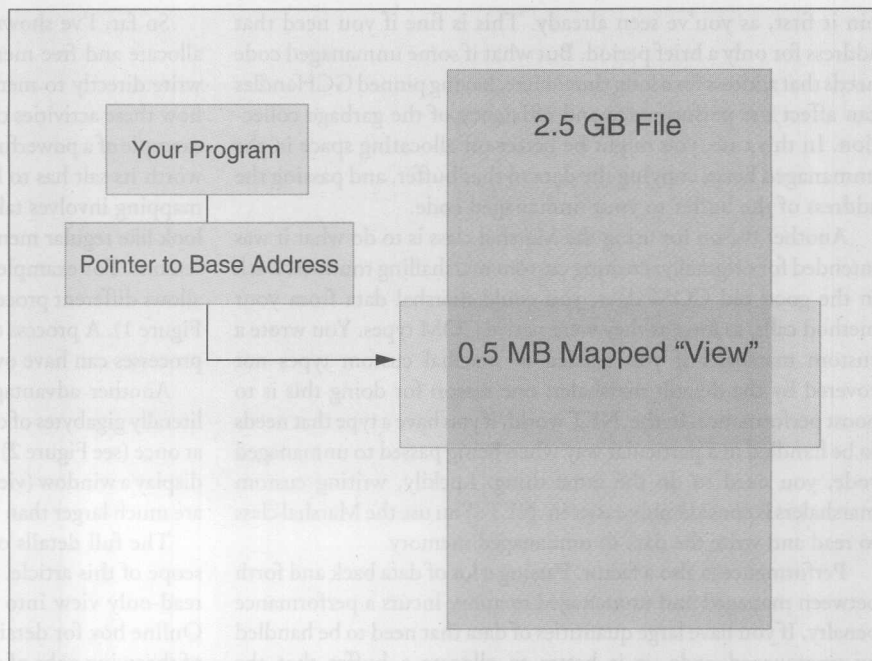
Dim BufferAddress As IntPtr = _
    Marshal.AllocHGlobal(256)
Marshal.Copy(MyData, 0, BufferAddress, 256)
```

To read an array from unmanaged memory, call the `Copy` method and pass the address of the buffer (IntPtr), the array you want to fill with the data from the buffer, the index of the array you want to start copying into, and the size of data you want to copy:

```
Dim MyData(255) As Byte
Marshal.Copy(BufferAddress, MyData, 0, 256)
```

The `Marshal` class has many more methods, but most of them are utilities that aid in interoperation with COM code.

I promised earlier that I would explain some of the uses for this type of direct memory manipulation, because it usually is not necessary if all you're doing is calling Windows API functions from .NET. One of the reasons is that to pass the address of a variable to unmanaged code, you need to








**Figure 2 Peer Into Files.** You can use memory-mapped files to get a small view of large files. This helps you avoid loading the entire file at the same time and can be useful for looking at one record or a page of records in an extremely large data flat file.

HTTP://

# www.mabry.com

**We Sell C++ Source Code for our Controls!**

Mabry's **Internet Client Pack/X** delivers Internet protocol components that are powerful, flexible, and easy-to-use.

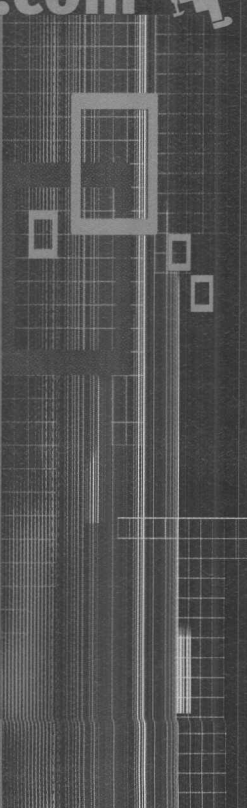
Every component is available in both ActiveX control and COM object forms.

**FUNCTIONALITY INCLUDES:**  
 POP, SMTP, FTP, HTTP, Sockets, RAS, UDP, Encoding, Decoding, Finger, Whois, MIME, UUencoding, DNS resolution, Time, News (NNTP), Ping, TraceRt.

Sample code included. C++ source code available.  
 Internet Client Pack/X \$349.00

Mabry Software Inc.  
 503 316th Street Northwest  
 Stanwood, WA 98292

Web: [www.mabry.com](http://www.mabry.com)  
 Email: [mabry@mabry.com](mailto:mabry@mabry.com)  
 Phone + Fax: 360.629.9278





pin it first, as you've seen already. This is fine if you need that address for only a brief period. But what if some unmanaged code needs that address for a long time? Here, having pinned GCHandles can affect the performance and efficiency of the garbage collection. In this case, you might be better off allocating space in the unmanaged heap, copying the data to that buffer, and passing the address of the buffer to your unmanaged code.

Another reason for using the Marshal class is to do what it was intended for originally: creating custom marshalling routines. Back in the good old COM days, you could marshal data from your method calls, as long as they were native COM types. You wrote a custom marshaler if you needed to marshal custom types not covered by the default marshaler; one reason for doing this is to boost performance. In the .NET world, if you have a type that needs to be handled in a particular way when being passed to unmanaged code, you need to do the same thing. Luckily, writing custom marshalers is considerably easier in .NET: You use the Marshal class to read and write the data to unmanaged memory.

Performance is also a factor. Passing a lot of data back and forth between managed and unmanaged memory incurs a performance penalty. If you have large quantities of data that need to be handled by unmanaged code, it is better to allocate a buffer that the unmanaged code would use and read/write smaller individual pieces of data to that buffer.

So far, I've shown you how to work with memory addresses, allocate and free memory on the unmanaged heap, and read and write directly to memory. I've even given some brief examples of how these activities can be useful. Now I'm going to show you an example of a powerful technique: file mapping. After all, any article worth its salt has to have a cool and practical example, right? File mapping involves taking a "view" of a physical file and making it look like regular memory in your process. This is useful for several reasons. For example, you can create a shared mapped file, which allows different processes to view the same "memory" contents (see Figure 1). A process can have multiple views of that file, and other processes can have overlapping views as well.

Another advantage of this technique: A process can look at literally gigabytes of data without having to load the entire contents at once (see Figure 2). Some "virtual" controls use this technique to display a window (view) of data to the user when the actual contents are much larger than would normally be acceptable to load at once.

The full details of memory mapping are far greater than the scope of this article, so I'll focus on showing you how to create a read-only view into a text file (download Listing 1; see the Go Online box for details). The first thing you need is a file. Instead of throwing gobs of code at you and making things really confusing, I offer a simple example: Copy a large text file and rename it (in this case, my example uses the filename "temp.tmp").

You open the file with a FileStream object and thereby eliminate the need to call the CreateFile API function. You need to do this to get a handle to the open file. Next, call the CreateFileMapping API function, which returns a FileMap handle. Now you're ready to create a view by calling the MapViewOfFile function, in which you specify the offset from the start of the file as well as the size of the view. The offset needs to coincide with the memory allocation granularity of the system. MapViewOfFile returns the base address of where the data will appear in "memory."

At this point, you can use that address along with the Marshal class methods I discussed in the previous section to read the data. If you had full memory mapped file capabilities in this example by creating it with read-write access, you'd be able to write to that "memory" too using the Marshal class members. In this example, I simply read the entire view as a string and display it to the console window. When you're done with a particular view, you must close the view using the UnmapViewOfFile function. When you're finished with the file, you must close the memory map file handle using the CloseHandle function.

With a little bit of extra work, you could

www.softelvdm.com

## Putting You In Control

Download a Demo Today!



### SftTree - Tree Control

Our tree control offers single, multiple selection, multiple columns, column resizing and reordering, ToolTips, FlyBy Highlighting, background bitmaps, cell editing, and many more grid-like features.



### SftMask - Masked Edit Control

Masked Edit control with user definable input mask, date and time input, drop down calendar, pop-up calculator, numeric fields, OLE drag & drop, cut/paste menu and many other features.



### SftBox - Combo Box

Our combo box supports multiple columns, headers, column reordering, font selection, auto-completion, auto-dropdown, cell bitmaps, OLE drag & drop and much more...



### SftTabs - Tab Control

Our tab control offers over 50 tab styles, including dual-sided tabs. Supports bitmaps, icons, fonts, disabled tabs, shared controls, ToolTips, complete design-time support and much more.



### SftButton - Button Control

This button control picks up where the standard button control leaves off. Multiple images, auto-repeat, toggle button, drop down button, focus control, multiline text, great customization features.

New Product!

Get All The Details!

Visit [www.softelvdm.com](http://www.softelvdm.com)!



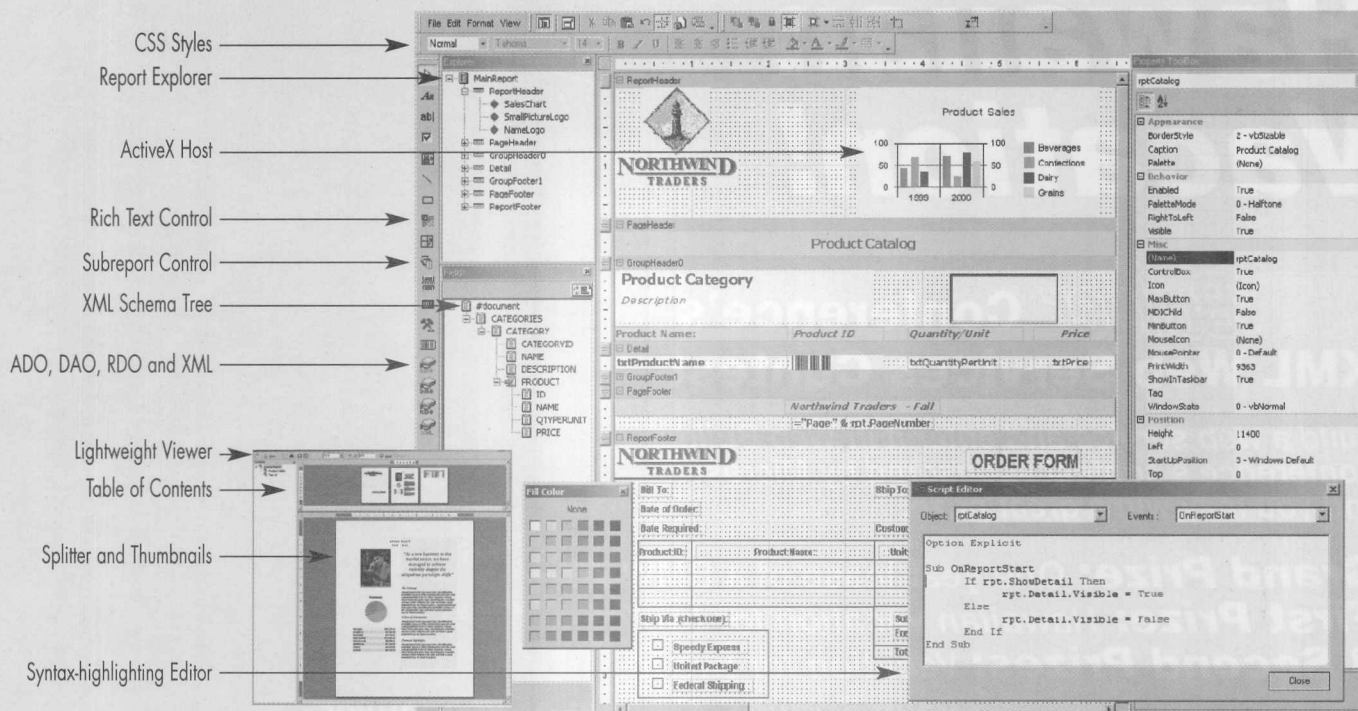
NOT THE INDUSTRY STANDARD

(941) 505-8600

FAX (941) 505-8555

**Active Reports 2.0** is an upgrade to our best-selling reporting component. Data Dynamics offers **Active Reports 2.0** in two editions to suit your development needs. Both editions build on the power and ease of ActiveReports 1.0 and add many new features and enhancements based on your feedback.

**Active Reports 2.0** Standard Edition now supports distributable XML report layouts, Active Scripting (VBScript/ JScript) events and expressions and XML data sources. **Active Reports 2.0** Professional Edition adds a royalty-free customizable end-user report designer control. Now your users can experience the power of ActiveReports within your applications.



## Data Access

- Supports DAO, RDO, ADO.
- New! Support for XML data sources and XPath, XSL expressions.
- Supports unbound data from arrays, collections and other data providers.
- Binds to recordsets and resultsets for maximum flexibility and connection efficiency.

## Architecture

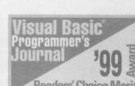
- Full runtime access to objects, data sources and binding, ability to add controls at runtime to modify the report behavior.
- Full control over printer jobs, properties and output devices.
- Hosts ActiveX components and OLE Objects.

## Performance

- Multi-threaded scalable engine.
- Fast response time.
- PRO! Report Caching service for fast web performance.

## Design Environment

- Easy to use Office-like designer environment.
- New! Access Reports import wizard.
- New! Crystal Reports import wizard.
- New! VBScript and JScript events and expressions and a built-in syntax-highlighting script editor.
- New! ActiveReports 1.0 upsizer.
- Includes data bound text, checkbox, image, OLE objects, rich text and subreports.
- ActiveX controls host. Use 3rd party controls or your own custom controls in your reports.
- New! Barcode control.
- New! Report bookmark and internet hyperlinks.
- New! Controls Style property editor based on standard Cascaded Stylesheets (CSS) syntax.
- PRO! Royalty-free end-user report designer control.



## Deployment

- Reports can be compiled for maximum security and speed.
- New! Report layouts can be distributed as XML-based report layout (RPX) files (application-host independent).
- Enhanced! Java viewer based on JDK 1.3 with printing support.
- New! TIFF export in addition to Text, RTF, Excel, PDF and HTML export filters.
- Enhanced! PDF export supports table of contents bookmarks and embedded fonts.

## ActiveX Viewer

- New! Multipage thumbnails.
- New! Split views.
- New! Ruler.
- New! Text search.

## Pricing

- Introductory pricing (expires July 31st):
  - Standard Edition: \$399
  - Professional Edition: \$999\*

\*(includes Royalty-free Report Designer)



# Win a BMW Lease or Hawaiian Vacation!

## **WEBBuilder** Conference's XML Web Services Contest

*Build a web service and enter it in Web Builder Conference's contest sponsored by Microsoft and you may win a great prize:*

**Grand Prize:** One year lease of a BMW Z3

**First Prize:** Hawaiian vacation for two

**3 Second Prizes:** Vacation for two to

*Palm Springs, Cancun or Vancouver*

For official contest rules and entry form, visit:

**[www.webservicescontest.com](http://www.webservicescontest.com)**

Sponsored by:

**Microsoft®**



**FTP** FAWCETTE  
TECHNICAL  
PUBLICATIONS

Microsoft is a registered trademark of Microsoft Corporation.  
WEB Builder is a trademark of Fawcette Technical Publications, Inc.  
XML is a trademark of MIT and a product of the World Wide Web Consortium.  
All other trademarks are property of their respective owners



make a robust class to handle memory-mapped files and even provide stream access.

## Delegate Your Callbacks

I'll mention one last .NET pointer feature: delegates, a powerful means of encapsulating function pointers. Much has been written on delegates, but I'll make a token mention here, because an article about pointers in .NET wouldn't be complete without them. Creating a delegate instance in VB is easy: You merely use the AddressOf operator. When a delegate is passed to unmanaged code, it is marshaled as an address. If you do pass a delegate to an unmanaged function, make sure you declare a delegate variable first and assign the value of the AddressOf operator to that variable, which you can then pass to the unmanaged code. This keeps the delegate in scope after the call to unmanaged code.

Delegates are interchangeable between functions, as long as the functions have the same signature—in other words, as long as they have the same number and types of parameters. Although the Delegate class is intended to be the primary mechanism for handling events in .NET, it also makes wiring up unmanaged callbacks simple. The biggest perk is that a delegate doesn't care if

your function is a static (Shared) function, as a VB6 module function would be, or if it's an instance method, like a VB6 class function. The benefit is that you can wire a callback to a class method simply by using a delegate variable with the result from an AddressOf expression. Matthew Curland wrote an excellent column on duplicating

this functionality for VB6 (see Resources), but it's a no-brainer in .NET.

The .NET architecture lets you write powerful managed code. However, if you need to reach beyond this to unmanaged code, the new tools I've covered in this article, in conjunction with PInvoke, should prove invaluable. **VSM**

**Robert Teixeira** is a Florida-based consultant who is assessing VB's ability to not only support the next generation of applications, but to lead the way to them. You can reach him at [robteixeira@msn.com](mailto:robteixeira@msn.com).

### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VB0110** Download all the code for this issue of VSM.

**VB0110RT** Download the code for this article separately. This article's code includes Listing 1, which shows you how to create a read-only view into a text file.

**VB0110RT\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

## Visual Studio® Programmers

### Develop apps for Palm OS®...

... and run those same apps on Pocket PC®

Visit the AppForge website today for details on our support for Color, Menus, and Pocket PC's. While you are there, get a free mobile development white paper and download a free evaluation version of AppForge Software.

Special Discounts available at [www.appforge.com/vs](http://www.appforge.com/vs)

[www.appforge.com](http://www.appforge.com)

Offer expires December 3, 2001.  
Visual Studio and Pocket PC are registered trademarks of Microsoft Corporation.  
Palm OS is a registered trademark of Palm, Inc. Copyright © 2001 AppForge, Inc.

**AppForge**  
because you're going places™

# Time for some serious KeyStone offers the most in-depth,



## Java™ GUI Programming

Build user interfaces using Swing  
Java Foundation Classes (JFC).

4 CDs

Single-User List Price

**\$599<sup>95</sup>**

\$699.00

## Java™ JDBC™ & Servlet Programming

Provides you a strong foundation in  
web development databases, SQL,  
HTML and Servlets.

4 CDs

Single-User List Price

**\$599<sup>95</sup>**

\$699.00

# computer training. self-paced training available.



## Visual Basic® 6.0

Learn API, ActiveX Controls, automation programming and much more.

17 CDs **\$1199<sup>95</sup>**  
Single-User List Price \$1869.00



## Java™ 2.0

Learn Java architecture, object creation, multi-threading capabilities, applets, & servlets.

8 CDs **\$599<sup>95</sup>**  
Single-User List Price \$879.00



## SQL Server® 2000

The entire course prepares you for Exams 70-228 and 70-229.

11 CDs **\$1099<sup>95</sup>**  
Single-User List Price \$1299.00



## Windows® 2000

Includes the learning courses related to exams 70-210, 70-215, 70-216, and 70-217.

25 CDs **\$1799<sup>95</sup>**  
Single-User List Price \$2750.00



## XML

Learn XML markup, content modeling, XML entities, scripting, dynamic sites and documents.

5 CDs **\$399<sup>95</sup>**  
Single-User List Price \$549.00



## Intro to Java™

Become proficient in the Java programming language. Create applets and applications.

4 CDs **\$599<sup>95</sup>**  
Single-User List Price \$699.95



## HTML & DHTML

Learn page attributes, legends, tables, padding attributes, HTML & DHTML structure & hyperlinks.

6 CDs, 3 Videos **\$699<sup>95</sup>**  
Single-User List Price \$989.00



## Oracle® 8iR2

This course will cover many of the complex topics needed for certification.

5 CDs **\$899<sup>95</sup>**  
Single-User List Price \$999.00

**10<sup>Year</sup>**  
**Anniversary**  
1991-2001

**Over 1200 Titles Available**



Please call for multi-user discounts and pricing. Prices shown are for single-user training only.

When it comes to self-paced computer training, no other company comes close. KeyStone packs the maximum amount of information in every course, making it easier to learn and retain. Exclusive KeyNotes keep it concise and come in a format you can review whenever you would like. With over 1200 self-paced training courses available, KeyStone Learning Systems has the learning course you need.

To Order or Receive A Free Catalog

**1.800.349.5761**

[www.keystonelearning.com](http://www.keystonelearning.com)

**KeyStone™**  
Learning Systems

Self-Paced Computer Training

A GlobalLearningSystems Company

CODE #MVP



# Enhance Performance With ADO Connection Objects

Use ADO Connection objects to achieve better performance and scalability in n-tier applications.

by Dimitrios Tsonis

## Technology Toolbox

- ☐ VB.NET
- ☐ C#
- ☒ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☒ VB6
- ☒ Other:
  - Microsoft ActiveX Data Objects (ADO) 2.6
  - Windows NT 4.0 plus Option Pack 4.0 (MTS) or Windows 2000

**M**anaging precious client and server resources efficiently is the key to making n-tier applications more scalable. Using ActiveX Data Objects (ADO) Connection objects can help your application respond faster and scale up more easily to hundreds of users in a corporate environment. In this article, I'll show you an efficient, flexible way to use ADO connections from Visual Basic. You'll create two projects, one COM-based and the other based on COM+ or Microsoft Transaction Server (MTS).

An ADO Connection object represents a unique session with a data source. In the case of a client/server database system, it might be equivalent to an actual network connection to the server. You'll take advantage of ADO's connection-pooling capabilities. Connection pooling enables an application to use a connection from a group of connections you don't need to reestablish for each use. Once you create a connection and place it in a pool, an application can reuse that connection without performing the complete

connection process. By using the technique presented in this article, you can allow many applications or business objects to share the connection parameters at the same time, offering centralized management of Connection objects.

To get started, think about how the client part of a typical two-tier client/server application works. When it requires data from a data source, a client application first opens an ADO connection, then opens a recordset against the connection and retrieves the desired data. The application then updates data or executes commands against the database by using the ADO connection. At any time—and depending on how many data forms a client has opened—you might have several active ADO connections to the same data source. Each connection consumes memory and occupies server resources.

This process is similar in a three-tier application. In the middle tier, the business objects need to retrieve data from the data source and pass it to the presentation objects. The presentation objects in turn manipulate the data and use the methods exposed by the business objects to update the data in the data source. In this

## Resources

- "Pooling in the Microsoft Data Access Components" by Leland Ahlbeck and Don Willits. Includes a link to download the MDAC Pooling Toolkit: <http://msdn.microsoft.com/library/techart/pooling2.htm>
- *Professional MTS MSMQ with VB and ASP* by Alex Homer and David Sussman (Wrox Press, 1998, ISBN: 1861001460)
- Microsoft Universal Data Access download page: [www.microsoft.com/data/download.htm](http://www.microsoft.com/data/download.htm)

## VB6 • Understand the MDAC Toolkit Approach

```
Dim globalCN As ADODB.Connection
Private Sub Form_Load()
```

```
Set globalCN = New ADODB.Connection
globalCN.Open "Provider=SQLOLEDB.1;" & _
    "Password=;Persist Security Info=True;" & _
    "UserID=sa;Initial Catalog=YourDB;" & _
    "Data Source=YOUR_SERVER"
```

```
End Sub
```

```
Private Sub Command2_Click()
Dim cn As ADODB.Connection
Dim dblStart As Double
Dim dblEnd As Double
dblStart = Timer
'You can do performance testing by changing
'the number of times the following loop is
'executed via an input textbox
For i = 1 To txtLoop.text
```

```
Set cn = New ADODB.Connection
cn.Open "Provider=SQLOLEDB.1;" & _
    "Password=;Persist Security Info=True;" & _
    "UserID=sa;Initial Catalog=YourDB;" & _
    "Data Source=YOUR_SERVER"
```

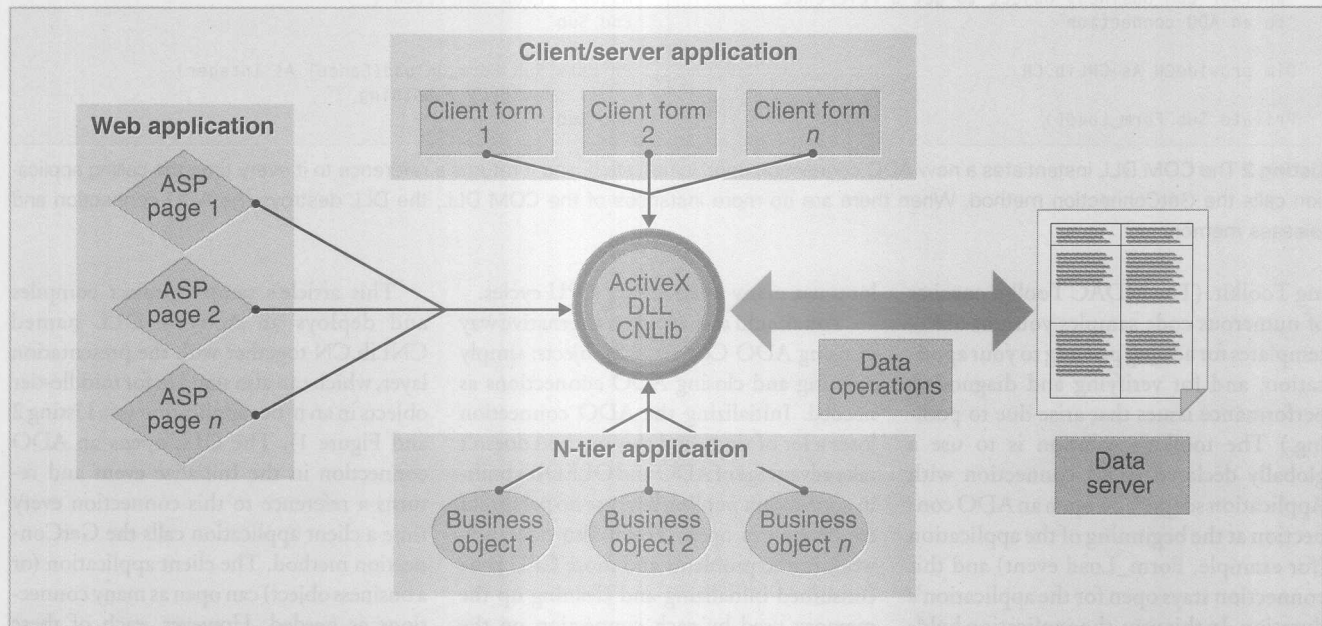
```
txtSample.Text = "Provider:" & _
    cn.Provider + Chr(13) & Chr(10) & _
    "Version:" & cn.Version
```

```
Set cn = Nothing
Next i
```

```
dblEnd = Timer
'display the result
txtTime.Text = Format(dblEnd - dblStart, _
    "0000.000") & " seconds"
MsgBox "Loop Completed !"
```

```
End Sub
```

**Listing 1** This sample test form demonstrates the MDAC Toolkit's approach. The form has a command button that, when clicked on, executes a loop through a textbox as many times as the user specifies. Each loop iteration opens a new ADO connection, retrieves the Provider and Version properties, and displays them in a textbox. When the loop code finishes executing, another textbox on the form displays the elapsed time.



**Figure 1 How the ActiveX DLL Works.** The ActiveX DLL you create in this article is the connection to all data operations requested by several types of applications. You can use the DLL from traditional client/server applications, n-tier applications, or ASP-based Web applications. The DLL provides a single source for holding the connection parameters to the database, and improves performance and increases scalability by reducing resource requirements.

case, the middle-tier objects open and manage all the required ADO connections. Usually, each middle-tier object opens and closes its own ADO connections as needed. If the presentation layer calls more than one method of a business object at the same time, the middle-tier object can open multiple connections simultaneously, resulting in a situation similar to the one described for a two-tier application: The application uses more memory than necessary and opens more resources than necessary on the data server end. Also, the application

replicates the connection-specific parameters (OLEDB Driver, User, Password, Database, and Server) inside all business objects, making it difficult to maintain the business objects in case users or developers need to change these parameters.

### Solve the Connection Problem

Microsoft offers one solution to this connection problem in its Microsoft Data Access Components (MDAC) Connection Pool-

## VB6 • Encapsulate an ADO Connection Inside a COM DLL

```

'compile the code below as an ActiveX DLL
'named CNLib.CN
Dim cn As ADODB.Connection
Private Sub Class_Initialize()

    Set cn = New ADODB.Connection
    cn.Open "Provider=SQLOLEDB.1;" & _
        "Password=;Persist Security Info=True;" & _
        "UserID=sa;Initial Catalog=YourDB;" & _
        "Data Source=YOUR_SERVER"

End Sub

Private Sub Class_Terminate()
    cn.Close
    Set cn = Nothing
End Sub

Public Function GetConnection() As _
    ADODB.Connection
    Set GetCN = cn
End Function

'example usage of CNLib.CN inside a VB form. Use
'a similar technique to use this object inside
'another COM business object to get a reference
'to an ADO connection

Dim provideCN As CNLib.CN

Private Sub Form_Load()

    Set provideCN = New CNLib.CN
End Sub

'Performance testing in a sample form
Private Sub CommandTest_Click()
    Dim cn As ADODB.Connection
    Dim dblStart As Double
    Dim dblEnd As Double
    dblStart = Timer
    'You can do performance testing by changing the
    'number of times the following loop is executed
    'via an input textbox
    For i = 1 To txtLoop.Text
        'set a reference to the connection provided
        'by the COM object
        Set cn = provideCN.GetConnection
        'display some attributes of the connection
        'object
        txtSample.Text = "Provider:" & cn.Provider & _
            Chr(13) & Chr(10) & "Version:" + cn.Version
    Next i
    dblEnd = Timer
    'display the result
    txtTime.Text = Format(dblEnd - dblStart, _
        "0000.000") & " seconds"
    MsgBox "Loop Completed !"
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set provideCN = Nothing
End Sub

```

**Listing 2** The COM DLL instantiates a new ADO connection upon initialization, and it returns a reference to it every time the calling application calls the GetConnection method. When there are no more instances of the COM DLL, the DLL destroys the ADO connection and releases memory.

ing Toolkit. (The MDAC Toolkit consists of numerous code samples you can use as templates for adding pooling to your application, and for verifying and diagnosing performance issues that arise due to pooling.) The toolkit's solution is to use a globally declared ADO connection with Application scope. You open an ADO connection at the beginning of the application (for example, Form\_Load event) and this connection stays open for the application's duration. In this way, the application holds an open connection for the length of its execution, regardless of whether it needs the connection. This keeps the ADO session pool active at all times (see Listing 1). However, this method doesn't do a good job of conserving memory and other resources at the client. Inside the OLEDB layer, the OLEDB connection pooling saves some server resources, but each open ADO connection on the client side consumes some memory. In addition, managing and cleaning up memory for all the connections opened and closed during the test

loop use many unnecessary CPU cycles.

You should also avoid an alternative way of using ADO Connection objects: simply opening and closing ADO connections as needed. Initializing the ADO connection loses a lot of time, and this method doesn't take advantage of ADO and OLEDB's built-in connection pooling because no persistent connection is open. You'll also have network traffic problems and more CPU time consumed initializing and cleaning up the memory used by each connection on the client side.

The best way to deal with connections is to create an ActiveX DLL (in-process server) that holds an open ADO connection and returns a reference to it as needed. This way to share an ADO connection between multiple business COM objects fits well within an n-tier architecture where you deploy the objects in the business-logic layer separately from the presentation objects. The presentation layer on the users' desktops doesn't consume any database resources this way.

This article's sample project compiles and deploys an ActiveX DLL named CNLib.CN together with the presentation layer, which can also stand in for middle-tier objects in an n-tier application (see Listing 2 and Figure 1). The DLL opens an ADO connection in the Initialize event and returns a reference to this connection every time a client application calls the GetConnection method. The client application (or a business object) can open as many connections as needed. However, each of these ADO Connection objects refers to an instance of the CNLib.CN you instantiated in the Form\_Load event.

I'll walk you through the code in more detail. First, you need to create the ActiveX DLL. Start by creating a new ActiveX DLL project in VB called CNLib and setting its property-threading model to be apartment-threaded. VB creates a class automatically; name it "cn." Set the class's Instancing property to Multiuse and the MTSTransactionMode property to NotAnMTSObject, because you won't be using MTS objects in



this example. Next, declare a variable inside the class that will serve as the ADO Connection provider to the calling applications:

```
Dim cn As ADODB.Connection
```

In the Class\_Initialize event, create an instance for the variable cn and open the ADO connection using various parameters. You might want to change the connection parameters to suit your needs or even use a Universal Data Link (UDL) file to provide these parameters:

```
Private Sub Class_Initialize()  
    Set cn = New ADODB.Connection  
    cn.Open "Provider=SQLOLEDB.1;" & _  
        Password=:Persist Security _  
        Info=True;" & UserID=sa; _  
        Initial Catalog=YourDB;" & _  
        Data Source=YOUR_SERVER"  
End Sub
```

You should also remember to close the ADO connection and release all resources in the Terminate event that executes when all the calling applications finish using the ActiveX DLL:

```
Private Sub Class_Terminate()  
    cn.Close  
    Set cn = Nothing  
End Sub
```

The only method offered by the ActiveX DLL returns a reference to the internal ADO Connection object. A calling application uses this reference in several ways I'll discuss later:

```
Public Function GetConnection() As _  
    ADODB.Connection  
    Set GetCN = cn  
End Function
```

## Test the DLL

You're ready to test the ActiveX DLL from a calling application after you compile it. Use VB to create a standard EXE project with one form that uses the ActiveX DLL. You need to include the CNLib DLL you just created in the new project's References section to use the ActiveX DLL object in your project. Then you can declare a variable from the CNLib DLL in your form:

```
Dim provideCN As CNLib.CN
```

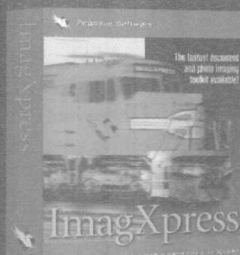
This variable is initialized in the Form\_Load event. At this point, the ADO connection inside the DLL is instantiated with these parameters:

```
Private Sub Form_Load()  
    Set provideCN = New CNLib.CN  
End Sub
```

Create a command button and several textboxes on the form to test the ActiveX DLL and its performance. Your goal is to simulate a full-scale application where ADO Connection objects are created and destroyed all the time while serving the data operations the application requires. So, program a For Loop that uses an ADO connection con-

# ImagXpress Professional

## Serious Imaging — Without Complications



Major corporations world-wide are making ImagXpress the cornerstone of their new and existing imaging projects. Why? Because ImagXpress gives developers imaging technology that is powerful, flexible and best of all reliable.

With over 80 image-processing functions and support for over 30 file formats, ImagXpress includes the standard imaging functions you expect, and the advanced imaging features you need, like super-fast deskew and despeckle. ImagXpress supports multi-page TIFF, including compacting of multi-page files, TIFF tag editing, and Wang TIFF annotation compatibility. ImagXpress includes TWAIN scanner support, video frame capture, advanced printing support, and fully-programmable annotations. ImagXpress also offers expandability with add-on toolkits for Barcode recognition and printing, Intelligent Character Recognition (ICR), Optical Mark Recognition (OMR) and high-speed ISIS scanning.

All of this comes in a clean, consistent programming interface across the various development environments you use. ImagXpress supports your platform of choice with COM objects, lightweight ActiveX controls, and Delphi VCLs and provides samples for Visual Basic, VC++, C++ Builder, Delphi, Access, and HTML.

Ease-of-use, high-performance, expandability, reliability and the best technical support in the business. These are the reasons why developers at major corporations are making ImagXpress their imaging toolkit of choice and that's why you should too. ImagXpress delivers serious imaging. Without compromise. Without complications.

Experience the power and flexibility of ImagXpress today by downloading a free unlimited-time trial toolkit at [www.pegasustools.com](http://www.pegasustools.com).



Pegasus Software  
4522 Spruce Street, Suite 200  
Tampa, Florida 33607

(813) 875-7575  
(800) 875-7009  
Fax (813) 875-7705

Free Trial

[www.pegasustools.com](http://www.pegasustools.com)

Bar Code • OCR  
Imaging

tinuously to retrieve certain parameters (you can substitute other data operations here, such as opening recordsets or calling stored procedures):

```
For i = 1 To txtLoop.Text
    'set a reference to the connection
    'provided by the COM object
    Set cn = provideCN.GetConnection
    'display some attributes of the
    'Connection object
    txtSample.Text = "Provider:" & _
        cn.Provider & Chr(13) & Chr(10) & _
        "Version:" & cn.Version
Next i
```

The value entered in a textbox on the form named txtLoop controls the number of iterations. Display the ADO Connection object's Provider and Version attributes in another textbox on the form named txtSample. During the For Loop, all ADO connections used through the variable cn point to the same Connection object in memory: the one instance of CNLib.CN you instantiated in the Form\_Load event. Having all open ADO connections point to the same Connection object in memory offers significant savings in memory on the client side and in resources on the server side. Think of an enterprise-wide application where the average user has three to four VB forms—each using at least one ADO connection to manipulate data—opened at the same time, and several hundred users are accessing the system concurrently. Following this method, this application could open from 300 to 2,000 fewer connections at the same time. This results in an application that can scale more easily and work well under heavy usage.

Another advantage of having all open ADO connections point to the same Connection object in memory: The connection parameters are located centrally inside one COM object. If any of these parameters changes, you don't need to recompile the entire application, which you'd have to do if you used the MDAC Pooling Toolkit's method. The method in this article takes advantage of the COM architecture in an easy-to-use way from VB. In the future, you can change the connection parameters inside the ActiveX DLL object. You only have to parameterize the ActiveX DLL object to deal with different connection parameters for different business objects if you need to use several different access permissions patterns.

## Use MTS or COM+

You can use MTS or COM+ to augment the performance advantages of using an ActiveX DLL to provide a reference to an open connection. You can use MTS' object-pooling capabilities to reduce resource overhead and enhance performance. Take advantage of MTS' object-management features by compiling the ActiveX DLL as an MTS object. You can give your object advanced security features and transactional capabilities, and you can deploy it easily in an enterprise-wide solution (download Listing 3 from the VSM Web site for an MTS implementation of the ADO Connection provider object; see the Go Online box for details).

The MTS code is similar to the COM code, but it has a few substantial differences. The ActiveX DLL implements the ObjectControl interface that's required for interaction with MTS.

It also requires you to declare a variable of typeObjectContext that's used for the stateless implementation:

```
Implements ObjectControl
Dim objObject As ObjectContext
```

Although the Class\_Initialize and Class\_Terminate events are the same as in Listing 2, you need to add two events related to the MTS implementation:

```
Private Sub ObjectControl_Activate()
    Set objObject = GetObjectContext()
End Sub
```

```
Private Sub ObjectControl_Deactivate()
    Set objObject = Nothing
End Sub
```

MTS calls the ObjectControl\_Activate event when the ActiveX DLL object is instantiated, and calls and the ObjectControl\_Deactivate event when the object is destroyed. In addition to these two events, add one more MTS-related method:

```
Private Function ObjectControl_CanBePooled() As _
    Boolean
    ObjectControl_CanBePooled = True
End Function
```

MTS calls the CanBePooled method to determine whether the ActiveX DLL supports object pooling. This is true in this case. The GetConnection function is also different from Listing 2:

```
Public Function GetConnection() As ADODB.Connection
    Set GetCN = cn
    'release context
    objObject.SetComplete
End Function
```

You call an additional line of code, objObject.SetComplete, to release the object context back to MTS. This allows MTS to make optimum use of resources.

An MTS implementation is most useful in a three-tier architecture where you deploy the business objects using MTS on an

## Understand Threading Models

A threading model is a set of rules describing the interaction of threading and COM that an object or a client follows. In the *apartment model*, the main rule is that a client application can call an object's methods only from the thread the object was created on.

In the *free-threading model*, a client application can call any object method or COM function from any thread at any time. It's up to the object to serialize access to all its methods to whatever extent it requires to keep incoming calls from conflicting. Free threading provides the maximum in performance and flexibility.

# Trust the LEADers...

More than 10 Years of Imaging Excellence

## Image Format Support

### RASTER

JPEG  
-Lossy  
-Lossless  
JPEG2000

MPEG

JBIG

PNG

SGI

PSD

PCD

EXF

FPX

EPS

PDF

Postscript

IOCA

MODCA

BMP

DIB

ICO

CUR

ANI

CLP

PCX

DCX

IMG

RAW FAX

WFX

MAC

TGA

RAS

PCT

CMP

OS/2 BMP

AWD

IFF

XPM

XBM

CUT

GIF\*

DICOM

CALS

WAN

WPG

WMF

EMF

AVI

FLC

MSP

RLE

PNM

PBM

PGM

PPM

XWD

TIFF 6.0

-Multipage

-JPEG

-Packbits

-Huffman

-RLE

-LZW

-CCITT

-CCITT G31D

-CCITT G32D

-CCITT G4

-RGB

-Palletized

-Grayscale

-Bitonal

-CMYK

-YCbCr

-CIE LAB

### VECTOR

WMF

EMF

DXF 2-D

DXF 3-D

DWF

DRW

DWG

CGM

DGN

HPGL

HPGL2

PICT

Gerber

VC++ • VB • C++ Builder • Delphi • .NET • API • C++ Class Library • ActiveX • VCL • COM

**LEAD Technologies** has more than ten years of experience developing digital imaging technology and creating programming toolkits for the development community. Over those years LEAD has created imaging technology that is currently used by Microsoft, Intel, Hewlett Packard, Eastman Kodak, Reuters, Corel and many other world class companies. Our experience developing imaging technology and solving programmer's problems have produced the engines listed below. Visit our web site for full details on available LEADTOOLS products which utilize these powerful engines.

## NEW!

### JPEG2000

J2K, JP2 Lossy or Lossless

### eMail

Add SMTP and POP3 functionality

### ePrint Multi-functional Printer Driver

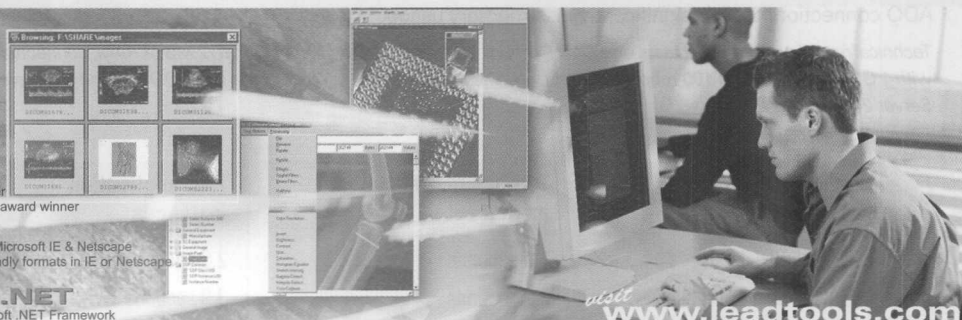
Includes Free SDK - a "Best of AIIIM 2001" award winner

### Browser Plug-in

Microsoft IE & Netscape  
Display, process and save non-Internet friendly formats in IE or Netscape

### LEADTOOLS for .NET

LEADTOOLS imaging toolkit for the Microsoft .NET Framework



## RASTER IMAGING ENGINE

**Import and export 60+ raster file formats** - Supports a wide range of compression options, bit depths (up to 64) and color spaces. Progressive and non-progressive modes, multi-page, animation and non-image data.

**Image processing** - Transforms- resize, rotate, linear and bicubic interpolation, flip, invert, reverse, crop, underlay, shear, transpose, auto deskew and combine. Filters - sharpen, intensity, saturation, histogram, posterize, median, edge, noise and more. Spatial filters - gradient, laplacian, sobel, prewitt and more.

**Display** - scroll, scale with interpolation, dither, contrast, brightness, with a choice of over 2,000 special effects.

Scanning, printing, imaging common dialogs, thumbnail browser, image list, database imaging functions, screen capture and much more!

## MEDICAL IMAGING ENGINE

**DICOM 3.0** - Supports the latest specs, including all standard IOD classes and modalities (CR, CT, MR, NM, US, RF, SC, VL, Worklist, etc.) and DICOM directory.

**DICOM communications protocol** - COMPLETE support including all Service Classes (Verification, Storage, Query/Retrieve, Patient Management, etc). High-level communications functions to simplify the creation of DICOM client/server applications.

**Optimized image processing** - The richest in the industry supporting 1,2,3,4,5,6,7,8,12,16,24 and 32 bit images. Includes 8-16 bit grayscale display with window leveling and LUT support.

Includes Medical specific measuring and mark-up annotations like cross product, ruler, point and protractor.

## INTERNET IMAGING ENGINE

**Support for client server development over the Internet or LAN** - Provides a framework for sending /receiving commands (LOAD, SAVE, CREATEWIN, etc.) from one computer to another.

**Control image processing remotely** - Allows for creation of "remote control" type applications where image processing can take place on a remote computer.

**Small COM objects and ActiveX controls** - Internet enable LEADTOOLS Raster imaging functionality but greatly reduce redistribution requirements.

**New HTTP and FTP functionality** - Provides programmatic control of FTP & HTTP servers.

**Upload Control** - For sending files to an HTTP server.

## 30-DAY MONEY BACK GUARANTEE (US & Canada)

\*License required for Unisys for formats using LZW compression. LEAD and LEADTOOLS are registered trademarks of LEAD Technologies, Inc. ISIS® is a registered trademark of Pixel Translations, a division of ActionPoint, Inc. All other product names are trademarks of their respective owners. LEADTOOLS is available in several versions, while not every feature is available in every version, you can easily find the toolkit to match your needs by visiting our website.

## DOCUMENT IMAGING ENGINE

**Powerful annotation capabilities** - Enable the addition of text, highlights, sticky notes, audio, ellipses, buttons, lines, arrows, rectangles, polygons, redaction, hotspots, freehand scribble, pointers, bitmaps, stamps, rulers and hyperlinks all with user-defined security features.

**Document image processing and clean-up** - Despeckle, deskew, inverted text, removal of dots, blobs, lines, borders, hole punches, character smooth. Region of interest, preview of changes and composite viewing of the modified regions.

**Optimized viewing of bitonal images** - Specialized display filters including FavorBlack and ScaleToGray, bilinear and bicubic interpolation.

Also includes ultra-fast CCITT G-3, G-4 and rotation, JBIG compression and a pan-window.

## VECTOR IMAGING ENGINE

**Import/export** - DWG, DXF, DWF, EMF, WMF, CGM, DGN, DRW, HPGL, HPGL2, PICT and LEAD VEC (LEAD's proprietary file format) in native form.

**Editing** - 14 different primitive object types (Vertex, line, rectangle, polyline, polybezier curve, polygon, ellipse, circle, arc, text, pie, chord, polydraw, and raster). Group objects into layers, copy or move objects between layers, lock individual layers. Add, edit, delete, rotate, translate and scale objects and layers. Convert points from world space to screen space and vice versa. Pixel accurate hit testing.

Also includes 3-D viewing (lighting, shadow, camera), 3 vector engines (GDI, OpenGL and DirectX), convert or overlay vector drawings to any LEAD supported raster format.

## MULTIMEDIA IMAGING ENGINE

**Play, Edit and Save** - Comprehensive support of multimedia formats including AVI, WAV, MIDI, SND, AIF, AIFC, MPEG-1, MPEG-2 files.

**Capture** - Capture multimedia data from any Video for Windows or DirectShow capture device.

# LEADTOOLS



**LEAD TECHNOLOGIES**  
INCORPORATED

1201 Greenwood Cliff, Suite 400  
Charlotte, NC 28204  
Call Toll Free: 1-800-637-4699

Sales: 704-332-5532 sales@leadtools.com  
Support: 704-372-9681 support@leadtools.com

1990-2000



Method	1,000 loop iterations	10,000 loop iterations	20,000 loop iterations
COM DLL CNLib	0.187 seconds	1.9 seconds	3.9 seconds
MDAC Toolkit approach	0.250 seconds	2.4 seconds	4.8 seconds

**Table 1 The COM DLL Method is the Most Responsive.** Here's a responsiveness comparison of the COM DLL you create in this article with Microsoft's MDAC Pooling Toolkit method. I tested 1,000, 10,000, and 20,000 loop iterations using the code in Listings 1 and 2. The COM DLL approach is 20 to 25 percent faster than the MDAC Pooling Toolkit method. Also, because the COM DLL returns a reference to an ADO connection, there's significantly less memory usage.

*Technical note: I performed all tests on a Dual Pentium III 500-MHz PC equipped with 512 MB of RAM, connected to a Dual Pentium III 600-MHz server equipped with 1 GB of RAM using a 100-mbps Ethernet LAN. The client ran NT 4.0 Workstation and ADO 2.6, and the server ran Windows 2000 Advanced Server and SQL Server 2000.*

Method	Two application instances	Four application instances	Five application instances
COM DLL CNLib	2.5 seconds	2.9 seconds	3.3 seconds
MDAC Toolkit approach	3.2 seconds	3.9 seconds	4.8 seconds

**Table 2 Examine the Methods' Scalability.** The response time changes when two, four, and five instances of the calling application are running at the same time and each instance goes through 10,000 loop iterations. The COM DLL method scales better than the MDAC Toolkit method. With two application instances opening and closing connections at the same time, the drop in performance is 31 percent for the COM DLL and 37 percent for the MDAC Toolkit approach. With four application instances, the COM DLL drops 52 percent and the MDAC Toolkit approach drops 63 percent in performance. With five application instances opened simultaneously, the COM DLL responds 73 percent more slowly, but the MDAC Toolkit method takes 100 percent more time to respond.

application server residing between the users' workstations and the data server. Again, imagine a corporate-wide solution where hundreds of users at once are accessing data and services. All business objects can use an instance of CNLibMTS.CN to get a reference to an open ADO connection. The resource savings on the application server and on the data server can be significant and allow the application to serve user requests more efficiently. Additionally, if the connection-specific parameters change, you need to modify only one object to make that change. Maintenance is much easier with this method.

### Create a Stateless Implementation

A proper implementation of an MTS object has to be stateless for MTS to reuse the instance of the component and associate it with a different context. A release of the object context (that is, stateless implementation) happens when you call the SetComplete or the SetAbort method inside the objects' methods. Although your implementation maintains some state, this state is always the same no matter how many applications are using the MTS object and calling its methods. The ADO Connection object opened in the Initialize event stays as is until your application terminates the last instance of the MTS object. Any application, business object, or data access object calling the MTS object's GetConnection method always receives the same result; the order of invocation doesn't matter.

Unfortunately, in VB6's implementation of the MTS object, you can't use the MTS object-pooling capabilities (derived through the "stateless" implementation) because the COM object is apartment-threaded and not free-threaded (see the sidebar, "Understand Threading Models"). However, you'll be able to do this in VB.NET when it's released. In the meantime, you can still take advantage of MTS' advanced security features and its transactional and centralized deployment capabilities.

Now, compare the DLL you've created with Microsoft's MDAC Pooling Toolkit method in terms of responsiveness, scalability, and ease of use, and you'll see that the COM DLL approach is the fastest and most scalable (see Tables 1 and 2). The COM DLL methods allow you to control connections centrally and let business-layer objects be independent of the data source being used. Each business object can get a reference to a connection opened by the COM DLL. In contrast, the MDAC Toolkit method hides the connection-specific parameters (OLEDB Driver, User, Password, Database, and Server) inside the application itself. Therefore, changing the data access methods results in changing the presentation objects, violating one of the primary principles of building n-tier applications.

In contrast, the COM DLL approach allows you to design an enterprise-wide application that makes more efficient use of valuable client and server resources, and at the same time allows changes to the data source without affecting either the presentation or the business-object layer.

You've now learned how to encapsulate an ADO connection within a COM object. Your data access objects, business objects, or presentation objects can share this COM object, making efficient use of one of the most important ADO resources. You can often apply the methods in real-life applications.

You might have an Active Server Pages (ASP) Web application, such as an e-commerce store or an online reporting system, in which several ASP pages need to access data from a data source. Have each ASP page use a connection provider COM object to get a reference to an active ADO Connection object. You can then use the ADO Connection object to execute commands such as stored procedures, retrieve data through recordsets, or pass it as a parameter to other business objects that accomplish data-related tasks.

This method consumes fewer resources while thousands of users access the ASP pages from the Internet. This is especially true if each

# has never been easier!

Windows  
**ActiveX**

**.NET** Framework

**Web** Components



## Communication Tools

Looking to foster a community on your Web site or intranet to improve employee knowledge, increase user-to-user interaction, and encourage customer feedback? No matter what your goals, ComponentOne has the ideal communication solution.

### • VSFORUM™ 2.0

Customizable discussion forums for Web sites or intranets

## Reporting Tools

You know the challenges of presenting and sharing critical enterprise information. Whether you require powerful, reliable reporting or need to preview, format, print, and export documents, rely on ComponentOne for the most flexible reporting solutions.

### • VSVIEW™ 7.0 Classic Edition

Document viewing, formatting, print previewing, and exporting

### • VSVIEW™ 7.0 Reporting Edition

VSVIEW Classic, plus Microsoft Access-style database reporting

### • True DBReports™ 6.0

High-performance reporting for Microsoft Visual Studio applications

## Documentation Tools

Need painless ways to produce Microsoft HTML Help, WinHelp, and standard HTML; automatically document ActiveX controls; and add spell-checking and thesaurus functionality to your apps? The search is over with ComponentOne's complete documentation solution.

### • True Help™ 1.0

The essential Help authoring tool for Microsoft Word

### • VSDOCX™ 1.0

Automatic ActiveX control documentation

### • VSPELL™ 6.0 Classic Edition

Spell-checker and thesaurus for Visual Basic applications

### • VSPELL™ 6.0 International Edition

VSPELL Classic, plus add-on British English, Spanish, and Dutch dictionaries

## .NET Framework Tools

Visit us for the latest information on upcoming ComponentOne tools for creating and using Web applications and services that take full advantage of the Microsoft .NET platform:

[www.componentone.com/net](http://www.componentone.com/net)

## Development Tools

Everything you need to develop database front-end applications, create and distribute dazzling charts, and rapidly build an OLE DB Provider. When you need robust, easy-to-use data access and presentation tools, rely on the ComponentOne development solution.

## Data Access Tools

### • ComponentOne Query™ 1.0

End-user ad hoc query builder

### • True DataControl™ 6.0

Enhanced data control for Microsoft Visual Studio

### • True OLE DB Provider™ SDK

The "build an OLE DB provider in one week, not one year" SDK

## Data Presentation Tools

### • True DBGrid® Pro 7.0

The enterprise grid

### • VSFlexGrid® Pro 7.0

The agile grid

### • True DBInput™ Pro 6.0

Data-aware ActiveX input controls for Microsoft Visual Studio

### • ComponentOne True DBList™ Pro 7.0

Supercharged DBList and DBCombo controls for Microsoft Visual Studio

### • ComponentOne Chart™ 7.0

The ultimate charting tool for distributed applications

### • ComponentOne WebChart™ 7.0

The ultimate charting tool for browser-independent Web server applications

### • ComponentOne SizerOne™ 7.0

Four-in-one resizing, tabbing, and parsing tool

### • Oletra® Resizer 2.0

Resizing control for code-free resolution independence

## ComponentOne tools cover the full range of your development needs!

Whether you need to get Visual Studio® and Web applications up and running in a flash, or will be making the move to the Microsoft® .NET™ Framework, ComponentOne makes the journey quick and easy.

No other company offers our complete series of high-performance tools for Visual Studio and Web development, our upcoming innovative products for Microsoft .NET Framework and emerging technologies, and our responsive team dedicated to providing top-notch service and support.

When you need to get from idea to application quickly and easily, the path is clear. Trust the performance, support, and value you can only get from ComponentOne tools. Download your eval copies today!

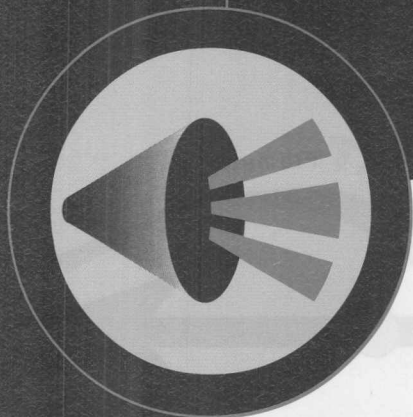
# ComponentOne

[www.componentone.com](http://www.componentone.com)

East Coast Office: 1.800.858.2739 or 1.412.681.4343 • West Coast Office: 1.888.228.4839 or 1.510.496.3240

ComponentOne™ LLC • 4516 Henry Street • Pittsburgh, PA 15213 • USA

© 2001 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.



# ANNOUNCING **.net** Magazine

the first and only .NET authority

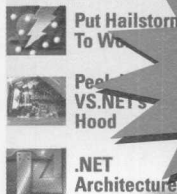
Make Your Web Services Secure

Fawcette Technical Publications, Fall/Winter 2001/2002, Vol 1 No.1  
[www.thedotnetmag.com](http://www.thedotnetmag.com)

## .net

Magazine

Exclusive!



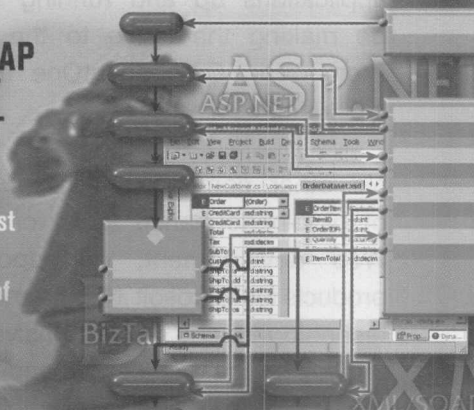
**FREE!**

## Smart Migration Strategies

- BizTalk
- XML/SOAP
- ASP.NET
- ADO.NET
- C#

Build Your First Web Service

Make Sense of .NET Server Strategy



Subscribe to the only publication devoted to helping you architect, build, deploy and maintain .NET-based applications and systems. You'll get how-to, hands-on information to help you:

- Learn smart strategies for migrating your servers to .NET
- Build and deploy your own Web services
- Integrate multiple .NET technologies into a powerful transaction tool
- Put the Hailstorm Web services to work
- Understand security and authentication in the brave new world of Web services

It's practical, timely and **FREE**.

Subscribe today: [www.fawcette.com/dotnet](http://www.fawcette.com/dotnet)

**SUBSCRIBE FREE!**



## [www.fawcette.com/dotnet](http://www.fawcette.com/dotnet)

All product names herein are the properties of their respective owners.

Publisher reserves the right to determine qualifications for free subscriptions. Inquire about rates for international subscriptions.



user opens several ASP pages at the same time. Compare this situation to the traditional approach of opening a separate ADO connection on every ASP page that accesses data. You can expect a significant improvement in response time using the shared COM object, achieving much greater scalability. You should implement the connection provider COM object as a Session object because application-level objects shouldn't be apartment-threaded. They should be both-threaded (that is, capable of apartment-threaded and free-threaded execution), while session-level objects can be apartment-threaded (download Listing 4).

You can also use your COM object in an n-tier application in which several business or data access objects need to access and manipulate data from a data source. Each object needs to open several recordsets, execute commands, and return results to the presentation-layer objects. To perform these tasks, all these objects need to open one or more ADO Connection objects to access and manipulate data from a data source. Assume you're deploying all these objects using MTS or COM+ on a particular object server, and also assume the presentation layer, which resides on the workstations of hundreds of corporate users, is using them. Compare the scenario of having all these objects use each of their own ADO Connection objects vs. sharing the one provided by the connection provider COM object. As in the preceding example, the system's performance and scalability increase greatly by using the connection provider COM object to provide a reference to an existing ADO

connection. Maintenance and change management are much easier because all the connection-specific parameters are located inside one object instead of being scattered inside several business or data access objects. **VSM**

**Dimitrios Tsonis** is the chief software architect at Berland Technologies Inc. in Culver City, Calif. His main focus is on designing and implementing highly scalable n-tier applications using Windows DNA technologies. You can reach him by e-mail at [dimitrios@berlandtech.com](mailto:dimitrios@berlandtech.com).

### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VB0110** Download all the code for this issue of VSM.

**VB0110DI** Download the code and listings for this article separately. This article's code includes the code to create the COM and MTS implementations of the DLL described in the article.

**VB0110DI\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).



**100%  
Satisfaction  
Guaranteed!**

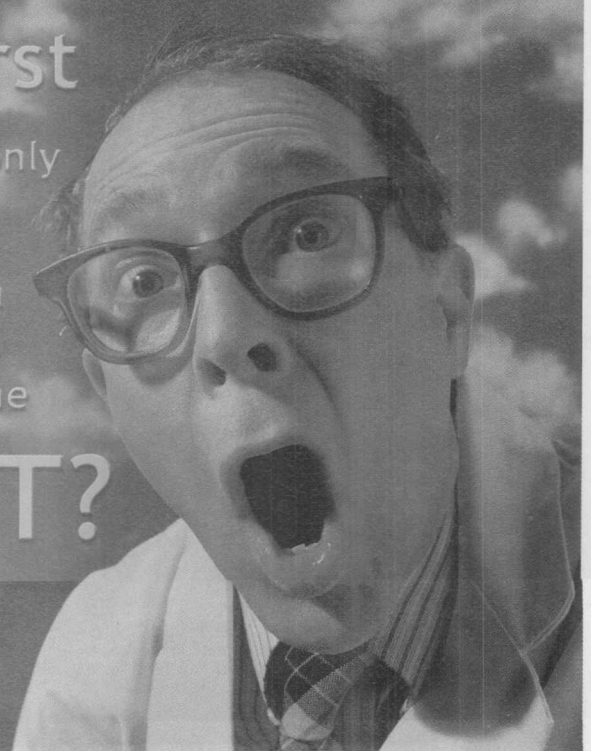


Ten years ago, CrypKey was the only commercial software-based copy protection. Today, it is still the best, with great features and reliability that come only from years of refinement.

CrypKey is easy to implement, easy to support, and offers more than just great protection, it provides lots of cool features that can simplify distribution and actually help you market your software!

- Use our SDK for complete control of your options & licensing, or CrypKey Instant to protect compiled code in minutes.
- Distribute via Internet - control & upsell features & options remotely.
- Single free trials convert to full versions. Unlike others, CrypKey is not fooled by back-dating, reinstalls or registry edits.
- Anti-hacker encryption technology included to enhance protection.
- License Transfer & superior Network capabilities.
- Using CrypKey and our e-commerce CASPER product, your software can accept credit card purchases and obtain its CrypKey authorization on-line, automatically, directly from your application - **with no browser required!** Users download, try & buy automatically!

Surprised  
that  
the First  
Software-only  
copy  
protection  
is also  
the  
**BEST?**



[www.crypkey.com](http://www.crypkey.com)

**Register Now!**  
800-848-5523  
[www.vcdc.com/vsm](http://www.vcdc.com/vsm)

# Visual C++

**Developers Conference**

**Orlando • October 9-14**

## **New Tools for Building Industrial Strength Apps with C# and C++.NET**

- ◆ **FULL BONUS DAY ON .NET!**
- ◆ **Get Face-to-Face with C#**
- ◆ **Learn the Ins and Outs of Migration to C# and Interop**
- ◆ **What the C# Enthusiast Needs to Know about VC++.NET**
- ◆ **Advanced ATL Drill Downs**
- ◆ **Dig Deep into Windows XP's**

**Call 800-848-5523 to Register**

**[www.vcdc.com/vsm](http://www.vcdc.com/vsm)**

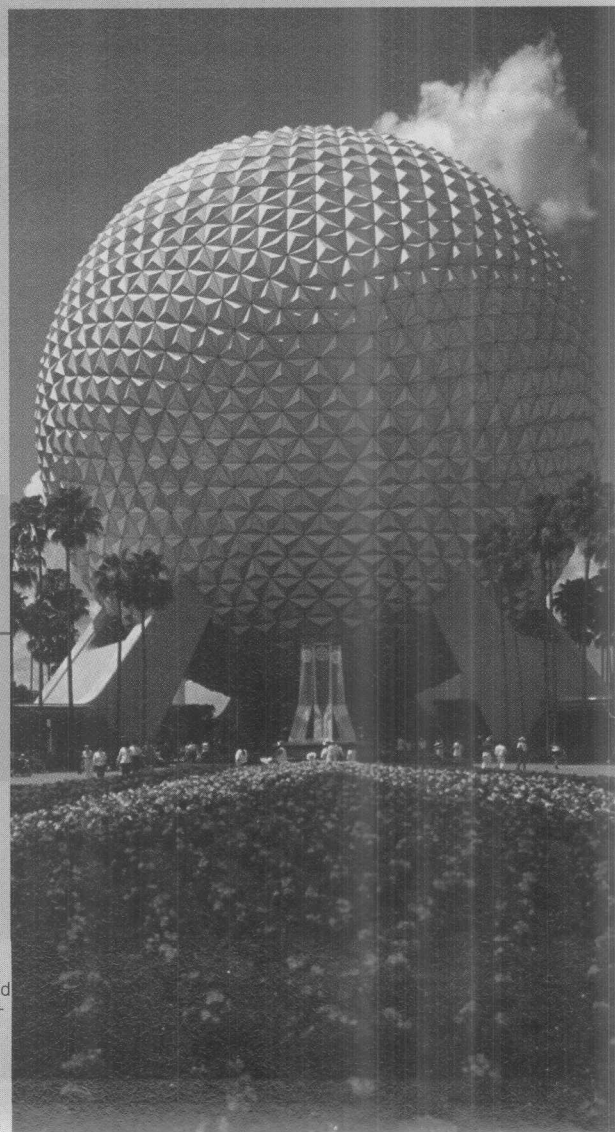
**FTP** FAWCETTE  
TECHNICAL  
PUBLICATIONS

**Visual Studio**  
Magazine

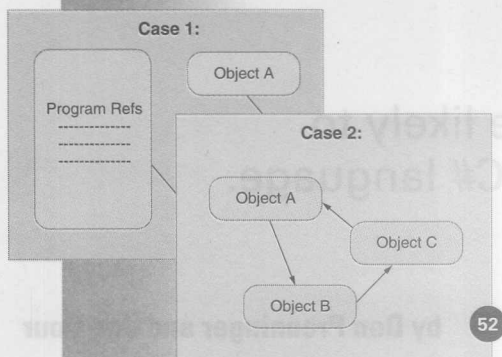
**Microsoft**

**msdn**

Microsoft, MS, MSDN, SQL Server, Visual Basic, Visual C++, Visual Studio, and Windows are registered trademarks of Microsoft Corporation. Visual Studio Magazine is a registered trademark and Visual C++ Developers Conference, is a trademark of Fawcette Technical Publications, Inc. Visual Studio is used by Fawcette Technical Publications, Inc. under license from Microsoft. All other trademarks are property of their respective owners.



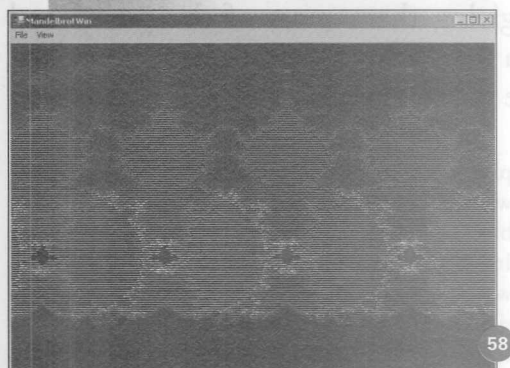




## 52 Clear Common C# Hurdles

Learn how to avoid some of the most common gotchas when working with C# from two developers who have been using the language for real development for more than a year and a half.

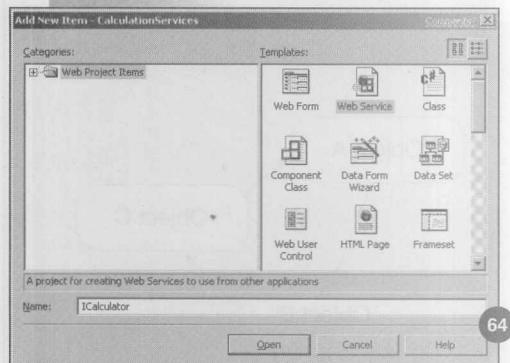
by Don Preuninger and Joe Dour



## 58 C# Explorer Manage C# Objects

Managing objects takes more than conserving memory. Take better advantage of the .NET garbage collector for efficient object management and increased performance.

by Bill Wagner



## 64 Web Services Develop Interface-Based .NET Web Services

Separating interface from implementation isn't just for traditional component-oriented apps—its benefits apply to Web Services too. Learn how to develop and consume interface-based Web Services in VS.NET.

by Juval Lowy



# Clear Common C# Hurdles

Avoid the most common gotchas you're likely to encounter when using Microsoft's new C# language.

by Don Preuninger and Joe Dour

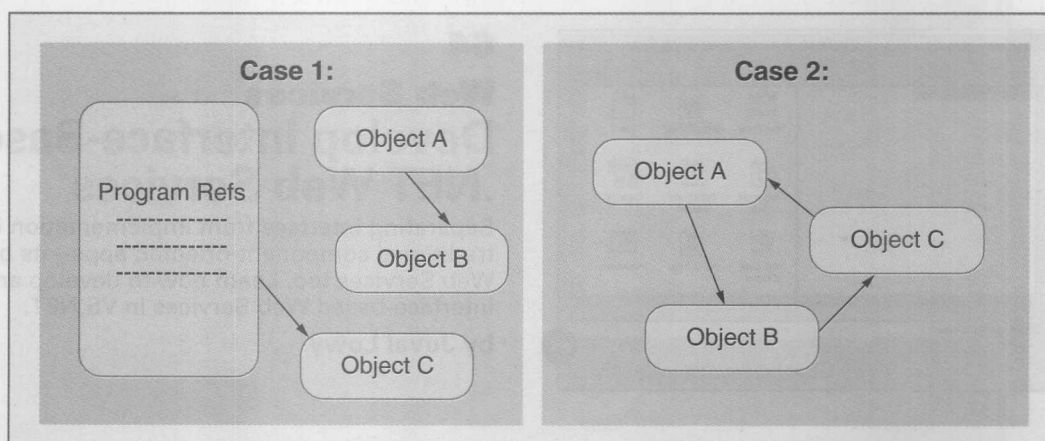
## Technology Toolbox

- ☐ VB.NET
- ☒ C#
- ☐ SQL Server 2000
- ☐ ASP .NET
- ☒ XML
- ☐ VB6

Nothing's perfect, and that includes programming languages. C# promises to simplify life for C++ developers, giving them most of the power and flexibility they're accustomed to, but with RAD advantages that have been part-and-parcel of the Visual Basic experience for years.

Microsoft has made significant progress with C#, but, as with any programming language, you as a developer can still encounter some significant issues and gotchas that pop up from time to time—issues that can give you headaches or indigestion if you're unprepared for them. At the same time, most of these issues are surmountable with a little forethought. We've been developing components for the .NET

platform for more than one and a half years now, and we've encountered a healthy number of stumbling blocks that have impeded our progress along the way. In this article, we'll share some of our insights on working with C#, including several of our more significant productivity enhancers. Your mileage will vary, of course, but we hope you find our set of tips as useful as we do.



**Figure 1 Collect the Garbage.** .NET objects are garbage-collected when they can no longer be reached from any executing code. In Case 1, Objects A and B are unreachable from the program, even though A maintains a reference to B. This means both will be garbage-collected. Object C is still reachable in Case 1, so it won't be garbage-collected until the reference to it goes out of scope. In Case 2, a circular reference exists between A, B, and C. In COM, these objects would never be destroyed because their reference counts would remain at 1 each. In .NET, they will be garbage-collected because they are unreachable from the executing program code.

## 1. Avoid the Tilde (~) Finalizer

Let's begin with a potentially confusing decision C#'s designers made: using the tilde (~) to denote a Finalizer. This can trip up C++ programmers, who naturally take this for a C++-style destructor, which has the same syntax but different functionality. Garbage collection is an article topic in itself, but you want to avoid using a tilde finalizer in .NET for several reasons. First, a finalizer is not deterministic, which means you have no guarantee when it will be called. Second, a finalizer carries additional overhead because the garbage collector must make at least two passes before it can collect objects with finalizers. Third, .NET's new garbage-collecting architecture calls a finalizer on a different thread, which means the code inside a finalizer must be thread-safe.

There's some good news, however. One major reason you need a destructor in C++—releasing references to other objects—goes away in .NET. For example, assume object A maintains a reference to object B when no other entity holds a reference to A or B (see Figure 1). Both will get garbage-collected (even though A still holds a reference to B). This is because the garbage collector flags both as nonreachable.

This also eliminates the circular reference bug, a common problem in COM development. Under .NET, if A holds a reference to B, which holds a reference to C, which holds a reference back to

A, and no reachable object holds a reference to any of these three objects, then A, B, and C will all be flagged as unreachable and eligible for garbage collection. In the COM world, which is based on reference counting instead of garbage collection, this situation results in the immortality of all three objects.

## 2. Clean Up Resources With IDisposable

You must still perform some cleanup when using .NET. Con-

### C# • Free Resources With the using Keyword

```
using System.Drawing;
class a
{
    public static void Main()
    {
        using (Font FontResource1 = new
            Font("Arial", 10.0f), FontResource2 = new
            Font("Arial", 10.0f))
        {
            try
            {
                // use Font Resources
            }
            catch (System.Exception e)
            {
                return;
            } // compiler will call Dispose on
            // FontResource1 and
            // FontResource2
        } // compiler will call Dispose on
        // FontResource1 and FontResource2

        Font FontResource3 = new Font("Arial", 10.0f);
        using (FontResource3)
        {
            // use FontResource3
        } // compiler will call Dispose on
        // FontResource3
    }
}
```

**Listing 1** C#'s `using` keyword enables the compiler to determine automatically when the `Dispose` method needs to be called to free resources. In this example, the `using` statement ensures the `Dispose` method of the font objects is always called, regardless of how the function is exited. The instantiated objects must implement the `System.IDisposable` interface in order for the `using` statement to do its job correctly.

### C# • Watch Out for Implicit Boxing

```
using System.Drawing;
using System.Diagnostics;
class test
{
    public static void Main()
    {
        int a = 0;
        int b = 0;

        ModifyTwoIntegers( ref a, ref b );

        Debug.WriteLine( a.ToString() );
        // will write out "5"
        Debug.WriteLine( b.ToString() );
        // will write out "10"

        a = 0;
        b = 0;

        // calling this method will cause an
        // implicit boxing operation making
        // copies of 'a' and 'b' which will
        // be discarded after the call

        ModifyTwoObjects( a, b );

        Debug.WriteLine( a.ToString() );
        // will write out "0"
        Debug.WriteLine( b.ToString() );
        // will write out "0"

    }
    public void ModifyTwoIntegers( ref int
        number1, ref int number2 )
    {
        number1 = 5;
        number2 = 10;

    }
    public void ModifyTwoObjects( object object1,
        object object2 )
    {
        object1 = 5;
        object2 = 10;

    }
}
```

**Listing 2** A boxing operation makes a copy of the value type and places that copy on the heap so it can be treated as a reference type. However, any changes made to the object inside the method aren't reflected back in the original value type object. For example, assume the values of `a` and `b` reflect the changes made after a call to `ModifyTwoIntegers`. Making a subsequent call to `ModifyTwoObjects` leaves the values of `a` and `b` unchanged because C# creates copies of the two integers through an implicit boxing operation and discards them after the call is made.

sider the case when an object holds some resources such as pens or database connections. In these cases, a destructor/finalizer is not appropriate because it's not deterministic. This means you could find yourself holding onto that resource for much longer than necessary.

Therefore, you need a deterministic way to free these resources. The .NET Framework defines an interface called `IDisposable` that provides a solution. If the class you create needs to free these kinds of resources, you should implement the `IDisposable` interface in it. `IDisposable` has a single method, `void Dispose()`. Place the code to free up the resources inside this method. Typically, you do this by calling some referenced objects' `Dispose` methods. All .NET Framework classes that require cleanup implement this interface, including the `Pen`, `Brush`, and `Font` classes. But be aware that someone still must call the `Dispose` method. People who use the class can do this explicitly or with the new C# *using* keyword.

#### C# • Avoid Implicit Boxing of Structs

```
using System.Drawing;
using System.Diagnostics;
class test
{
    public static void Main()
    {
        Foo a = new Foo();

        // call Bar thru the interface
        ((IFoo)a).Bar();

        // this will write out 'false' because the
        // above call performed a boxing
        // operation that created a
        // copy of the Foo struct, which was
        // discarded immediately after the call
        Debug.WriteLine( a.BarCalled.ToString() );

        // call Bar directly on the struct
        a.Bar();

        // this will write out 'true' as expected
        Debug.WriteLine( a.BarCalled.ToString() );
    }

    public interface IFoo
    {
        void Bar();
    }

    public struct Foo : IFoo
    {
        public bool BarCalled;

        public void Bar()
        {
            this.BarCalled = true;
        }
    }
}
```

**Listing 3** Calling the `Bar` method by casting the struct to the `IFoo` interface causes an implicit boxing of the struct. This creates a copy of the struct on the heap—a copy that is discarded immediately after the call. This is not only inefficient, but it means any state changes made inside the method call aren't reflected in the original struct.

### 3. Handle Objects That Consume Resources With the *using* Keyword

The C# language includes the *using* keyword, which enables resources to be freed automatically for specified objects, regardless of how or when the objects go out of scope (see Listing 1).

The *using* keyword provides a cleaner, more efficient way of disposing resources than the traditional approach, where you must perform resource cleanup explicitly at each point of exit. You can ensure your applications don't leak resources by always adhering to the policy of working with the *using* keyword instead of calling the `Dispose()` method explicitly.

### 4. Beware of Implicit Boxing

.NET includes two types of objects: value types and reference types. Value types are classes derived from `System.ValueType` and include the intrinsic types such as `int`, `bool`, `short`, and so on. All other classes are reference types. The one exception to this rule is a struct, which is a value type. These types differ in where they are allocated and how they are referenced. You can create value types only on the stack (inside a method) or inline (as a member of a class or struct). You always create reference types on the heap, where the garbage collector manages them.

This is a major departure from C++. Classes and structs in C++ are almost identical, and you can create any object (including intrinsic types) on the stack or on the heap. Also, the only way to get a reference to a value type in C# is as a parameter in a method marked "ref." If you pass a value type as a parameter into a method that expects a reference type parameter, the C# compiler performs an implicit boxing operation (see Listing 2).

### 5. Don't Implement Interfaces on Structs

C# allows structs to derive from one or more interfaces. This might sound like a nice capability, but it carries certain risks that can produce unwanted results.

Structs are value types, so you can create them only on the stack or inline (as a member of a class). They are efficient because the garbage collector doesn't need to manage them. Unfortunately, this means the only way you can get a reference to a struct is by passing it as a ref argument into a method.

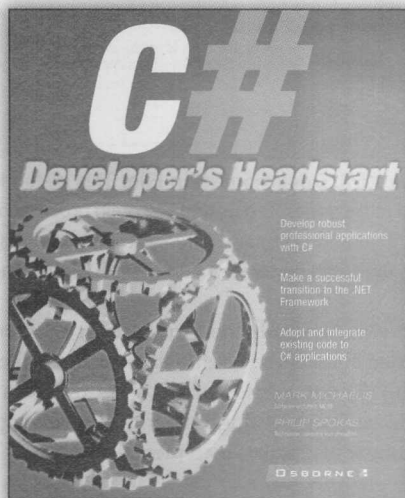
You're faced with a question: How can structs implement interfaces, which are, in effect, references? The answer: When you access the struct by casting it to an interface, an implicit boxing operation takes place and a copy of the struct is placed on the heap. You end up with a reference to the copy. When you call a method on the interface that changes the state of the object, you actually change the state of the copy, which gets discarded after the call (see Listing 3).

If you have a struct that needs to implement an interface, consider using one of two approaches. The first: Declare it as a class instead of a struct. Declaring it as a class means the object is no longer creatable as an efficient stack object. On the other hand, you avoid implicit boxing when accessing the interface. The second approach: Keep your struct a struct, but create a wrapper class that contains the struct as a public member. Implement the interface off the class instead of the struct. You can then use the struct as a stack-based object or instantiate the wrapper class when the interface is required. You're actually performing the equivalent of a boxing operation, but it will be explicit rather than implicit, ensuring you won't do so accidentally.

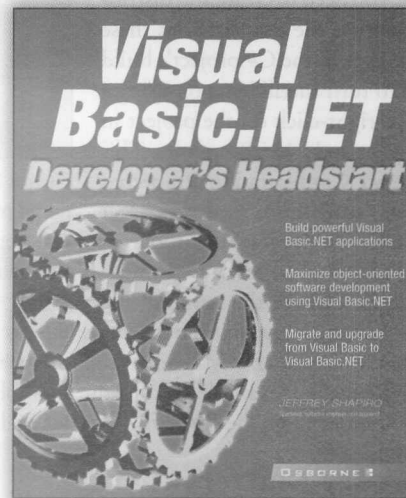


# Get Ahead with Headstart

Get a head start on Microsoft's new technologies—C# and Visual Basic.NET. Osborne is ahead of the curve on both counts with two new Developer's Headstart guides. *C# Developer's Headstart* and *Visual Basic.NET Developer's Headstart* are thorough, insightful developer's resources designed to fill your head with useful knowledge on the topics that matter to you.



**C# Developer's Headstart**  
0-07-219116-3  
M. MICHAELIS & P. SPOKAS  
\$24.99



**Visual Basic.NET Developer's Headstart**  
0-07-219581-9  
J. SHAPIRO  
\$24.99

## ALSO AVAILABLE



**Instant SQL Server 2000 Applications**  
G. BUCZEK  
0-07-213320-1  
\$49.99



**FrontPage 2002 Developer's Guide**  
M. JENNETT  
0-07-213259-0  
\$59.99



**Building Kylix Applications**  
C. JENSEN & L. ANDERSON  
0-07-212947-6  
\$59.99

**AVAILABLE NOW!**

**Free sample chapters available at [Osborne.com](http://Osborne.com)**

Available at bookstores and online retailers everywhere.

**Mc  
Graw  
Hill**

**OSBORNE**  
[www.osborne.com](http://www.osborne.com)

## 6. Enable XML Documentation Generation

Typing a triple slash (`///`) in the line before any element—whether a class, struct, interface, or any other element—prompts C#'s editor to generate a set of XML tags. You can place documentation for the element inside these generated XML tags. IntelliSense displays this documentation when you type in code that references the member.

Getting the benefit of this IntelliSense documentation feature is complicated if the generated assembly is a class or control library that will be used in other assemblies. In this case, you need to create and distribute an XML documentation file that contains these tags. You can do this as part of the C# project's build process, but you need to turn IntelliSense on using a four-step process. First, right-click on the project in the Solution Explorer pane and select Properties. This brings up a dialog; select Configuration Properties from the left-side tree. Next, make sure you select Build under Configuration Properties. Finally, the list on the right includes an item called XML Documentation File (see Figure 2). Specify the name of the XML file in this space. You must give this file the same name as the corresponding assembly, but with an XML extension. Otherwise, IntelliSense can't find the file. For example, assume the assembly is named `Foo.Bar.dll`; you must name the XML file `Foo.Bar.xml`.

## 7. Reduce Memory Clutter

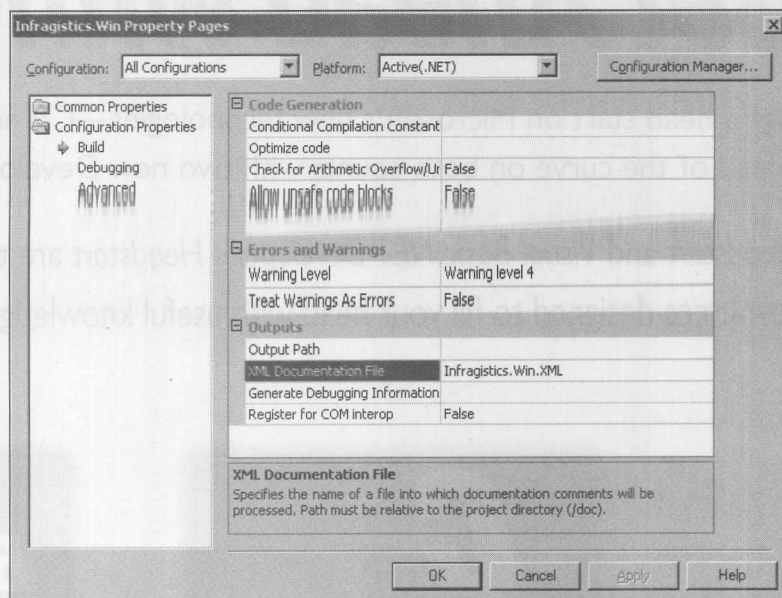
Instances of the `String` class are immutable objects in C#. This means you cannot change a string object once you create it. It also means the system must create new string objects to incorporate changes any time you manipulate instances of string classes. For example, assume you need to compose several date components to yield a long date format:

```
string FullDateTime;
string DayOfWeek = "Tuesday";
string Month = "March";
int DayOfMonth = 22;
int Year = 2001;

FullDateTime = DayOfWeek + ", " + Month + " " +
DayOfMonth.ToString() + " " + Year.ToString();
```

In this case, you create a new string object each time the `+` operator is used to increment the length of the overall result. In the process, you create and throw away five strings before the final sixth string is allocated and assigned to the `FullDateTime` variable.

Wasting a few string objects might not be significant in this example, but consider what happens when a method that performs extensive text manipulation does so by using the convenient addition operator. For example, assume you have a Web application that generates a complex string of HTML to be sent down to a browser. There might be hundreds of statements that concatenate various strings before writing them to an `HttpTextWriter` object. Each time the HTML string is extended, the app creates a new string to hold the longer string, and the old string is thrown onto the garbage heap. Over an extended period of time, such activity gradually depletes



**Figure 2 Specify the XML Documentation File.** This dialog lets you specify the name of the XML file to be generated during the build process of a C# project. Note that you set the Configuration dropdown to All Configurations, which causes the XML documentation file to be generated during both debug and release builds.

memory and degrades performance on a server where the page is requested frequently. Eventually, garbage collection will run and clear up the old, unused strings—but there is a better way.

You can use the `StringBuilder` class to build up and extend strings that consist of smaller, incremental units:

```
StringBuilder buffer = new StringBuilder();
buffer.Append(DayOfWeek);
buffer.Append(", ");
buffer.Append(Month);
buffer.Append(" ");
buffer.Append(DayOfMonth.ToString());
buffer.Append(" ");
buffer.Append(Year.ToString());
```

The `StringBuilder` class utilizes a single buffer, which is capable of expanding to whatever size is required. This second example requires writing a bit more code, but it doesn't change the size or performance of the compiled IL. Using the `StringBuilder` class instead of performing string concatenations saves you considerable memory usage and helps you avoid fragmentation.

## 8. Use *is* and *as* Keywords for Type Checking

Strong type checking is a basic aspect of the .NET Common Language Runtime, so sometimes you must verify the type of an object in an application before calling one of its methods or properties. Failure to do this leads to an unhandled exception, and the program will terminate if you attempt to access an object as a type other than what it is. This situation can occur when an ambiguous object is passed to a method and the method must determine the object type prior to using it. You can do this in several ways in C#.

For example, assume you need a method that determines the type of object passed as a parameter and only processes the object if

it's of type string or of type int. You can use exception handling to check an object's type in this circumstance (see Listing 4).

The exception-handling approach works as expected, but it's somewhat clumsy and awkward to use. An effective alternative: Take advantage of C#'s *is* keyword to accomplish the same goal with considerably less code and more readability. This code produces the same output as the first example:

```
public int foo(object o)
{
    if(o is string)
        return processStringObject(o);
    else
        if(o is int)
```

```
        return processIntObject(o);
    else
        return -1;
}
```

You have yet another option available to you. You can use the *as* keyword to make the object assignment, assigning a null value if the object is not a compatible type. One limitation of the *as* keyword: It can be applied only to variables that are object references; it can't be applied to value types. This is because you cannot assign value types a null value, which is crucial if you want to test the result of the object conversion.

So, this code uses the *as* keyword in the test for the string type (a reference type), but it uses the *is* keyword to test for the int type (a value type):

```
public int foo(object o) {
    string s = o as string;
    if(s != null)
        return processStringObject(s);
    else
        if(o is int)
            return processIntObject((int)o);
        else
            return -1;
}
```

This version of *foo* also produces the same output as the first example.

This demonstrates that you can often use C#'s *is* and *as* keywords interchangeably. Both keywords can be useful, and both are much more readable and understandable than the use of exception handling in these situations. **VSM**

### C# • Determine the Type of Object

```
class Class1 {
    static void Main(string[] args) {
        int result;
        ProcessObjects process = new ProcessObjects();

        result = process.foo("hello");
        Console.WriteLine("The first call returns {0}", result);

        result = process.foo(10);
        Console.WriteLine("The second call returns {0}", result);

        result = process.foo(35.3);
        Console.WriteLine("The third call returns {0}", result);
    }
}

class ProcessObjects
{
    public int foo(object o) {
        try {
            string s = (string)o;
            return processStringObject(s);
        }
        catch (System.InvalidCastException) {
            try {
                int i = (int)o;
                return processIntObject(i);
            }
            catch (System.InvalidCastException) {
                return -1;
            }
        }
    }

    public int processStringObject(string s) {
        return s.Length;
    }

    public int processIntObject(int i) {
        return i.ToString().Length;
    }
}

// The output from this sample:
// The first call returns 5
// The second call returns 2
// The third call returns -1
```

**Listing 4** You can use exception handling to determine the type of an object. Simply "try" different objects to see if you have a match. If the object type cast fails, an exception is thrown, and you try another type of object in the catch block.

**Don Preuninger**, vice president and senior architect at Infragistics, was a cofounder of ProtoView Development Corp. He created and developed many of ProtoView's early products for Windows 2.0 using C and C++, and he's been instrumental in the design and implementation of the forthcoming Infragistics line of .NET components. Reach Don by e-mail at [donp@infragistics.com](mailto:donp@infragistics.com).

**Joe Dour**, also a vice president and senior architect at Infragistics, was a cofounder of Sheridan Software and has spent more than 25 years designing and developing software. He too has helped design and implement the forthcoming line of Infragistics' .NET components. Reach Joe by e-mail at [joed@infragistics.com](mailto:joed@infragistics.com).

### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of VSM.

**VS0110DP** Download the listings for this article separately.

**VS0110DP.T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).



# Manage C# Objects

.NET has a garbage collector, but managing objects takes more than conserving memory. Here's how to manage object lifetimes effectively.

by Bill Wagner

## Technology Toolbox

- ☐ VB.NET
- ☒ C#
- ☐ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☐ VB6
- ☒ Other:

Visual Studio .NET  
beta 2 or later

C# programs run in a managed environment, and that environment manages memory resources for you. Even with such memory-management features, however, you still need to address programming issues that encompass your objects' lifetimes. You need to consider how long objects live and how the runtime environment destroys these objects. You might also need to manage more closely the resources objects acquire on your behalf. Even using pointers in unsafe code has ramifications for the garbage collector. In this article, I'll show you how to work with the runtime's garbage collector when you use .NET classes and how to write classes that work efficiently in the managed environment (see the sidebar, "Object Management Guidelines").

Along the way, I'll add animation to an application that displays the Mandelbrot set, a collection of complex numbers that creates fractal patterns when graphed. In tuning the application for performance, you'll learn more about how to use C# to work with the memory resources in the Common Language Runtime (CLR).

Essentially, the garbage collector frees and compacts memory by removing unnecessary objects, freeing you from dealing with the memory used by objects you allocate. However, you can structure your code to take better advantage of the garbage collector and increase performance.

## Resources

- *A Programmer's Introduction to C#* by Eric Gunnerson [Apress, 2001, ISBN: 1893115623]
- *Inside C#* by Tom Archer [Microsoft Press, 2001, ISBN: 0735612889]
- *Chaos: Making a New Science* by James Gleick [Penguin, 1988 (reprint edition), ISBN: 0140092501]
- "Garbage Collection: Automatic Memory Management in the Microsoft .NET Framework" by Jeffrey Richter [MSDN]: <http://msdn.microsoft.com/msdnmag/issues/1100/gci/gci.asp>
- "Garbage Collection—Part 2: Automatic Memory Management in the Microsoft .NET Framework" by Jeffrey Richter [MSDN]: <http://msdn.microsoft.com/msdnmag/issues/1200/gci2/gci2.asp>
- Visual Studio .NET home page: <http://msdn.microsoft.com/vstudio/nextgen/default.asp>

The .NET Runtime Garbage Collector uses a "Mark and Compact" algorithm that, instead of searching for dormant objects, finds and marks objects still in use. Any objects not marked as "in use" are freed, and the heap is compacted. Live objects move in memory when the garbage collector runs. The garbage collector runs only when necessary, such as when an allocation fails. The more objects you allocate, the more often it runs.

The garbage collector makes some assumptions about the likelihood of objects being unused: If an object survives garbage collection, it will likely survive future garbage collections. The idea is that local variables do not survive their first garbage collection. Any variable that does survive its first garbage collector operation is probably owned by an application-level object and therefore likely to be around for a while. If an object survives two or more garbage collector operations, the application probably owns it, so will likely be around for a long time.

The garbage collector first examines the zero-generation objects—objects that have never been examined. If the application needs more free memory, the garbage collector examines the first-generation objects and, if necessary, the second-generation objects. The goal is to free enough memory from the garbage collector operation with minimal work.

Your mission is to avoid creating extra objects needlessly and avoid calling the garbage collector too frequently. The garbage collector is efficient, however, and you want to work *with* it, not against it. You don't need to avoid creating objects for fear of garbage collection, but you do need to avoid being wasteful; if you can reuse an object as easily as creating a new one, do so.

For example, consider the differences between the `String` and `StringBuilder` classes. Examine this code from my Mandelbrot application:

```
String msg = "This shows the \
    Mandelbrot Set from the Range:";
msg += "\nx=(" + minX.ToString()
    + ", " + maxX.ToString() + ")";
msg += "\ny=(" + minY.ToString()
    + ", " + maxY.ToString() + ")";
```

```
msg += "\nEach Pixel represents (" +
    scaleX.ToString() + "," +
    scaleY.ToString() + ")";
```

String is a different animal. Each time the msg string is assigned a new value, a new string is created. You can write this more efficiently using the StringBuilder class:

```
StringBuilder msg = new StringBuilder
    ("This shows the Mandelbrot Set \
    from the Range:");
msg.AppendFormat (" \nx={0:e4}, \
    {1:e4} \ny=({2:e4}, {3:e4})", \
    minX, maxX, minY, maxY);
msg.AppendFormat (" \nEach Pixel \
    represents ({0:e4},{1:e4})", \
    scaleX, scaleY);
System.Windows.Forms.MessageBox.Show \
    (msg.ToString());
```

Only one String is built in the second example, located in the msg.ToString() call that displays the message. In the first example, 19 different String objects are created—one for each string literal, one for each ToString() call, and one each time the variable msg is assigned. So 19 different objects are orphaned after this code executes. Although each object is small, you can see how this can add up over time.

## Dispose of Objects

How do you free nonmemory resources? Well, you must do it yourself. Any object owning resources that must be freed must implement the IDisposable interface. After finishing with the object, you call Dispose() on the object, which frees any resources it has acquired (such as file handles, locks, cursors, and open database tables).

Whenever you use objects, determine whether they implement the IDisposable interface. If so, call the Dispose() method when you finish using the object. This is the most efficient way to free nonmemory resources. If you forget to call Dispose(), however, the Finalize() method found in most objects takes care of resources. (I'll explain more about how Finalize() affects performance later.)

If you create a class that acquires resources that must be freed, you should implement the IDisposable interface for that class so clients can call the Dispose() method to release resources. To implement the interface, declare that your class supports the IDisposable interface and implement the Dispose() method.

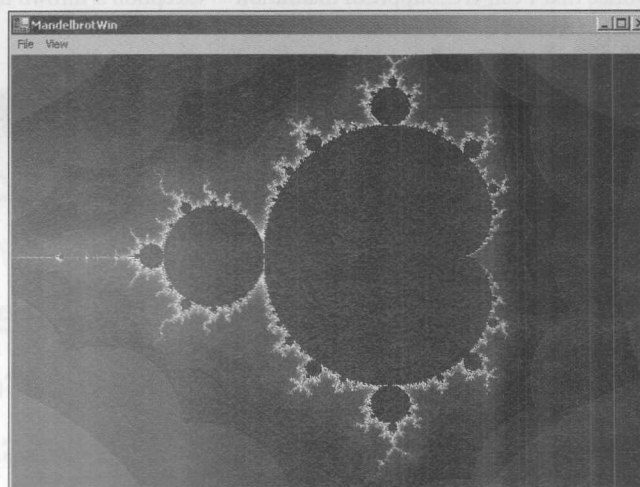
Dispose() and Finalize() work together often. Suppose you have a class that needs a handle to some resource. This code shows how the Dispose() method would work:

```
class MyClass : IDisposable
{
    private Handle hnd;

    MyClass ()
    {
        hnd = new Handle ();
        hnd.Acquire ();
    }

    public void Dispose ()
    {
        hnd.Release ();
        hnd = null;
        GC.SuppressFinalization \
            (this);
    }
}
```

```
}
public void Dispose ()
{
    hnd.Release ();
    hnd = null;
    GC.SuppressFinalization \
        (this);
}
```



**Figure 1 Animate the Set.** Although you can't see it on paper, this window is cycling through all the colors in the set in real time. You can see the interesting effects for yourself by downloading the code.

## Object Management Guidelines

**F**ollow these guidelines for working with the Common Language Runtime (CLR):

1. Let the garbage collector do its work. Avoid using Finalize() or Dispose() to release memory resources.
2. Finalize() slows down the garbage collector. If you implement Finalize(), you should also implement Dispose() so you can clean up the object more effectively.
3. Use the GC.SuppressFinalization() method to remove disposed objects from the finalization list.
4. Make sure the Dispose() and Finalize() methods can be called in any order and more than once.
5. Test for an already disposed object in public methods, at least in debug builds.
6. Use the IDisposable interface if an object supports it. If possible, use the "using" statement to ensure exception-safe code.
7. Use weak references for large objects that might be needed later. This technique can improve performance greatly, but you should use it sparingly—only when you're working with large, resource-intensive objects. Using it too often takes more cycles to resurrect objects.
8. If you use unsafe code to boost performance, make sure you don't keep unsafe pointers around long because they prevent the garbage collector from doing all its work. You need to check unsafe code carefully and use it only when you have profiled the code you want to change.

```

}

protected void Finalize ()
{
    hnd.Release ();
}
}

```

Notice that I call `GC.SuppressFinalization` (this) to remove the object from the finalization queue (I'll describe this later when I delve into finalization). You can also access an object that has been disposed. Using other methods, you would need to check that `hnd` isn't null and reacquire the object if necessary (this is why the `hnd` variable is set to null in the `Dispose()` method).

You don't need the `IDisposable` interface for memory resources—in fact, you *shouldn't* use it. Instead, use the `Dispose()` method to free other system resources, such as files, database tables, or other resources.

## Exceptions and Resources

The techniques I've covered so far are pretty simple. When you finish using an object, the garbage collector handles the memory re-

sources for you. If the class supports the `IDisposable` interface, you call the `Dispose()` method when you're done with the object. But what happens when exceptions occur? Because `Dispose()` is not called automatically during the stack unwinding process, you need to structure your code so you call `Dispose()` if your function exits normally, or if an exception is thrown through your functions.

**You don't have to implement a finalizer when you use the `Dispose()` method.**

You can do this in one of two ways.

First, you could add your own *try* and *finally* blocks to ensure `Dispose()` gets called. Suppose `myClass` is a class that implements the `IDisposable` interface. You could write this code:

```

myClass obj = new myClass ();
try {
    obj.Function ();
}
finally
{
    if (null != obj)
        ((IDisposable)obj).Dispose ();
}

```

C#'s *using* statement, however, gives you a much easier way. The statement indicates that you're using a variable inside a block and that the object should be disposed of when the block exits. You can replace the previous example with this code:

```

using (C obj = new C ())
{
    obj.Function ();
}

```

# INSTANTLY SEARCH GIGABYTES OF TEXT

**dtSearch®** The Smart Choice for Text Retrieval® since 1991

## Fast, precision searching

- ♦ over two dozen text search options
- ♦ most indexed searches take less than a second, even through very large databases
- ♦ also has unindexed searching

## Organization-wide reach

- ♦ automatically recognizes word processor, database, spreadsheet, email, PDF, ZIP, HTML, XML, Unicode files and more
- ♦ FindPlus® distributed searching extends the reach of a single search request to remote enterprise servers
- ♦ point and click setup

## Hit highlighting

- ♦ highlights hits in HTML and PDF while keeping embedded links and images intact
- ♦ converts other file types to HTML for display with highlighted hits



VISIT  
[www.dtsearch.com](http://www.dtsearch.com)  
 FOR 30-DAY  
 EVALUATION VERSIONS  
**1-800-IT-FINDS**  
[www.dtsearch.com](http://www.dtsearch.com)  
[sales@dtsearch.com](mailto:sales@dtsearch.com)



**WEB**  
**dtSearch Web**  
 Add instant searching to your site. See demo at [www.dtsearch2.com](http://www.dtsearch2.com)  
 ♦ \$999 per server



**SPIDER**  
**dtSearch Spider**  
 Spider and search Web sites ♦ included with all dtSearch products



**ENGINE**  
**dtSearch Text Retrieval Engine**  
 Add power searching to a product ♦ extensive sample source code in multiple programming languages ♦ from \$999



**DESKTOP**  
**dtSearch Desktop**  
 Find anything, anywhere, instantly ♦ \$199



**NETWORK**  
**dtSearch Network**  
 Search the many forms of data that exist across a large enterprise network ♦ from \$800



**PUBLISH**  
**dtSearch Publish**  
 Publish a searchable database to CD, DVD, etc. ♦ from \$2,500

**"Superb ... a multitude of high-end features"**  
 —PC Magazine

**"A powerful text mining engine ... effective because of the level of intelligence it displays"** —PC AI

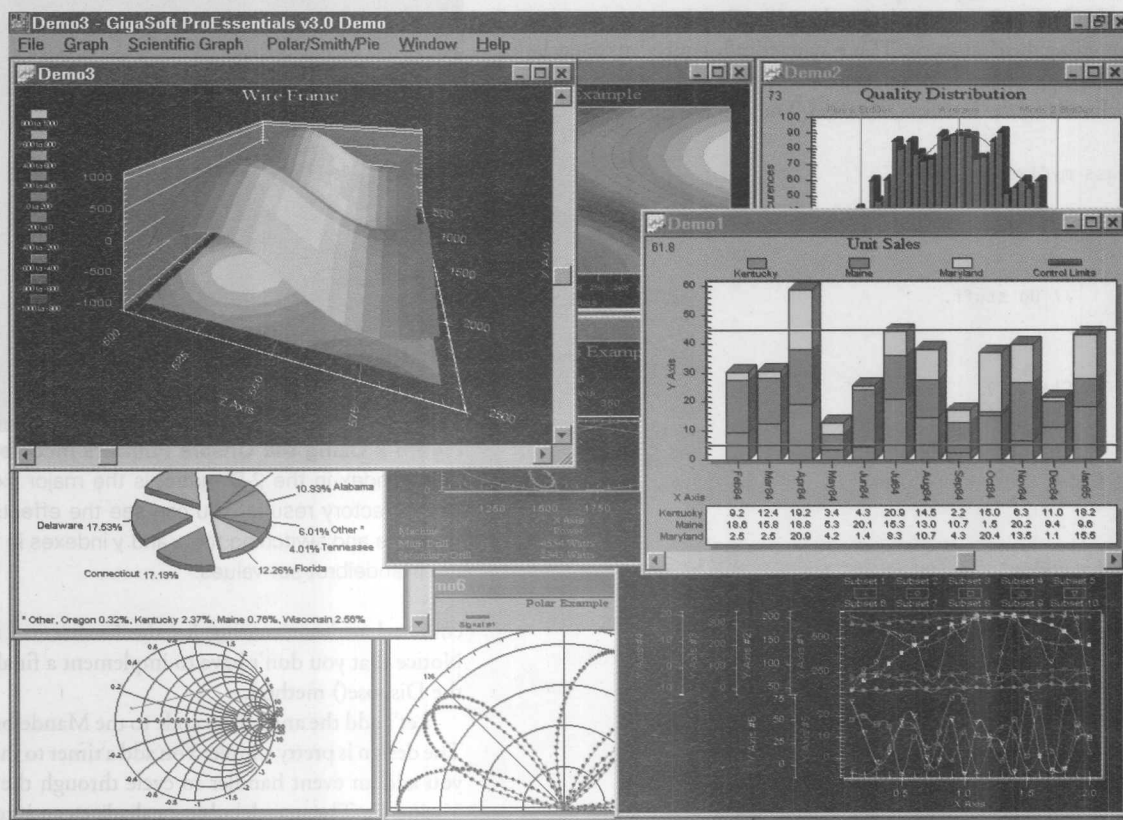
**"Very powerful ... a staggering number of ways to search"**  
 —Windows Magazine

**"Impressive"**  
 —PC Magazine Online

**"A tremendously powerful and capable text search engine"**  
 —Visual Developer



Whether you have important users, process serious data, or just want to produce the most impressive software possible, nothing will serve your needs or users better than ProEssentials.



## Why should you use ProEssentials?

1. ProEssentials provides the most features, implemented in the highest quality, and accessed via the cleanest programmatic interface.
2. Gigasoft provides professional technical support that is free, fast, and helpful. You will always have access to ProEssentials' developers.
3. ProEssentials is industry proven producing a continuous flood of positive customer feedback. Most of our customers have tried the competition and feel ProEssentials is "superior by far."
4. Gigasoft is dedicated toward customer-service and customer-orientation. With the help of our customers, Gigasoft will define and develop the future of quality charting functionality. Come join us in developing your charting tool.

Version 3 adds over 150 new features in addition to an already rich feature-set. Version 3 also adds significant improvements to its 580 page user's manual and online help. It also adds ten times more example code simultaneously written in C, VB, Delphi, and Builder flavors. ProEssentials provides many unique features found only in ProEssentials, like up to twelve automatic y axes and dual x axes in up to six sub-graphs, built-in zooming (even across axes), data-cursors providing data-point focus, images prepared and cached in memory for smooth flicker-free scrolling and real-time, no overlapping text, a synchronized presentation style grid, extensive easy to use annotation mechanism, and intelligent date/time handling for continuous as well as discontinuous time data. There's also standard features like drill-down, customization dialogs, popup-menus, and quality export capabilities including JPEG.

ProEssentials is ideally suited for technical, financial, data-acquisition, and any application needing the ultimate in clean professional appeal. Why settle for the average charting tool plagued with overlapping and illegible text, slow flashing redraws, and disappointing images/performance due to realistic amounts of data. Instead, ProEssentials is specifically engineered to be unsurpassed at quickly and clearly communicating your application's real-world data as well as being easy to implement and maintain.

\$549 includes five controls: Graph, Scientific Graph, 3D Scientific Graph, Polar/Smith, and Pie Chart objects. It also includes 32 and 16 bit OXC, VCL, DLL, and VBX interfaces for VB, Delphi, VC, Builder and others. It also includes free technical support, royalty-free distribution, and a 90 day total satisfaction guarantee.

Gigasoft, Inc.  
206 Bear Hollow Keller, TX 76248  
phone: (817) 431-8470  
fax: (817) 431-9860  
email: info@gigasoft.com

Demo and Trial-Edition at  
**www.gigasoft.com**

**Delphi**  
INFORMANT MAGAZINE  
RUNNER UP BEST CHARTING/MAPPING TOOL  
READERS CHOICE 2000  
ProEssentials  
Gigasoft

The `Dispose()` method is called at the close of the using block, whether or not any exceptions are thrown.

In C#, destructors are declared just like C++ destructors: with a “~” preceding the class name. This is syntactically equivalent to declaring a `Finalize()` method. In fact, the C# compiler replaces destructor calls with calls to the `Finalize()` method. You can use this code:

```
class myClass
{
    public myClass ()
    {
        // Do stuff.
    }

    ~myClass ()
    {
        // Free stuff.
    }
}
```

or this code:

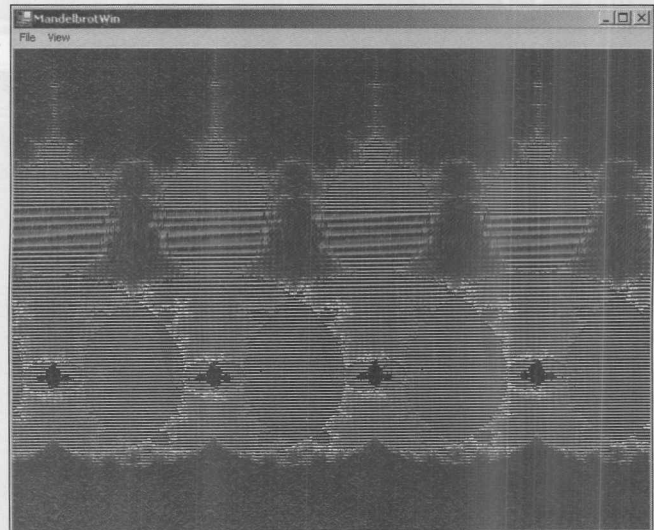
```
class myClass
{
    public myClass ()
    {
        // Do stuff.
    }

    protected override void Finalize()
    {
        // Free stuff.
        base.Finalize ();
    }
}
```

Whichever you use is a matter of style; I would expect C++ programmers to use the destructor syntax, while those with Visual Basic and Java backgrounds might be partial to the `Finalize()` syntax. Remember: The `Finalize()` method is *not* called when an object goes out of scope, as it is in C++. Instead, it is called as part of the garbage collector operation. For that reason, I prefer the `Finalize()` syntax to the destructor syntax; it looks different because it *is* different. Finalization has performance implications as well. Let's revisit the discussion of the CLR garbage collector from the beginning of this article and look at what happens when objects to be released have finalizers.

The runtime keeps track of objects that require finalization. When the garbage collector runs, it moves any objects that could be collected but require finalization to a “ready for finalization” list. After garbage collection, the finalizer thread wakes up and calls the `Finalize()` method on the objects on the finalizer list. The objects are removed from the finalization list and are now ready for the next round of garbage collection. Objects that require finalizers slow down the system and live longer than necessary when garbage collections occur.

To improve this situation, implement the `Finalize()` method, the `IDisposable` interface, and the `Dispose()` method. In your `Dispose()` method, call `GC.SuppressFinalization (this)`, which removes the object from the finalization queue. Now the object can be garbage-



**Figure 2 Using the Unsafe Pointers Incorrectly.** If you forget which index in the 2-D matrix is the major axis, you get some unsatisfactory results. You can see the effects by downloading the code and switching the x and y indexes in the matrix storing the Mandelbrot set values.

collected without waiting for the finalization thread to execute. Notice that you don't have to implement a finalizer when you use the `Dispose()` method.

Let's add the animation code to the Mandelbrot set application. The design is pretty simple: You add a timer to the application, then you add an event handler to cycle through the colors (download Listing 1). The mandel object calculates an integer value for each (x,y) point in the Mandelbrot set. The colormap is a class in the project that transforms a grayscale value into a color value to create color images. (Download the code for the entire example to see the finished version; see the Go Online box for details.) The `Dispose()` method in Listing 1 lets the other components release their resources. The new version of the application object does not need to have a `Finalize()` method; the timer's own `Finalize()` frees any resources if for some reason `Dispose()` is not called.

## Use Weak References

Now I'll address performance issues in the Mandelbrot application because the first version of the animation code is way too slow. One way to boost its speed is to avoid recalculating the Mandelbrot set points every time the colors change when the set gets animated: You add a class (download Listing 2) that holds the number of iterations in the Mandelbrot set. The class simply stores a matrix of integers; the matrix is the same size as the display. Now, when the set is animated, you need to change only the pixels. You don't need to recalculate the set points each time the display changes, saving more than 301,000 calculations. However, the application now uses much more memory, even when you're not animating the set. You can save this cost by replacing strong references with weak references.

A weak reference is a reference to an object that can be garbage-collected—essentially, it lets you set aside an object in case the system needs to collect it. If you don't access that object before a garbage collection occurs, the garbage collector removes it from memory. If you do access the object before a garbage collection, the object is returned. Saving the Mandelbrot matrix as a weak reference keeps the cached results in memory if memory is not becoming scarce (download Listing 3). The advantage is that you do not need to re-create the



image to animate it. In addition, if memory becomes scarce and the system needs to perform a garbage collection, you don't need to do any extra work—the system frees the reference for you.

The application performed better with the weak reference, but it's still not fast enough. To improve the speed further, I used unsafe code and pointers. I added the `Display()` method to the `PelMap` class (download Listing 4). This routine grabs a pointer to the start of the matrix of integers and a pointer to the start of the bitmap image. Then I use pointer arithmetic to move through both blocks of memory. The `unsafe` keyword marks the block as not running inside the managed environment. The two fixed blocks specify pointers to raw memory. The `fixed` keyword notifies the runtime that the object being pointed to cannot move in memory. Without the fixed blocks, the runtime can move the object in memory as part of a garbage collection action, which is one of the main reasons for the performance improvement.

With managed references, you must resolve each access independently because the object might have moved between instructions. Also notice that I reversed the *x* and *y* indexes in the cache. The first index in the matrix is incremented last, and that must be the *y* index. The `Turnerizer` transforms the integer values into color values for display. Now the animation routine runs in real time. Download the code and try it yourself. You can even zoom into different areas of the set while it's animated. Try switching the *x* and *y* indexes in the cache to get some visually interesting effects (see Figures 1 and 2).

I've shown you how to work with the garbage collector to

improve your application's performance. Most of the time, you'll be better off letting the garbage collector do its thing without any particular guidance. When you write code that has some unique memory usage characteristics, however, you can use the techniques I've described to modify the default behavior. **VSM**

**Bill Wagner** is a founder and software consultant for SRT Solutions, and he has more than 15 years of software development experience. He has been a columnist for *Visual C++ Developers Journal* and has written two eBooks on STL programming, available at [www.mightywords.com](http://www.mightywords.com). Contact Bill at [wwagner@srtsolutions.com](mailto:wwagner@srtsolutions.com).

### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110CE** Download the code for this article separately. This article's code includes Listings 1 through 4 and the finished project that creates, displays, and animates the Mandelbrot set.

**VS0110CE\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

**Microsoft**  
CERTIFIED  
Partner

## Not all Windows Installer tools are created equal.

*As you begin traveling down the Windows® Installer highway, make sure you have the power you need to reach your destination.*

Other installation tools may get you started using Windows Installer technology, but only one has the power you need, both today and down the road. Wise for Windows Installer offers both a visual, step-by-step approach to creating installs as well as the ability to view and edit any Windows Installer table for complete control. No other product offers this level of flexibility.

With additional features such as leading edge .NET support, the most powerful MSI debugger available, and the ability to integrate your installation with your automated software build process, Wise for Windows Installer is the most powerful and complete Windows Installer tool on the road. Don't get left behind.

From Wise Solutions, the expert in Windows Installer technology. [www.wisesolutions.com/wisewin11.htm](http://www.wisesolutions.com/wisewin11.htm)



Software installations made easy™



# Develop Interface-Based .NET Web Services

Building interface-based Web Services lets you switch between different service providers with minimal or no changes to client code.

by Juval Lowy

## Technology Toolbox

- ☐ VB.NET
- ☒ C#
- ☐ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☐ VB6
- ☒ Other:

Visual Studio .NET  
beta 2 or later

Windows 2000 or  
Windows XP beta 2  
or later

## Resources

MSDN for Visual Studio .NET

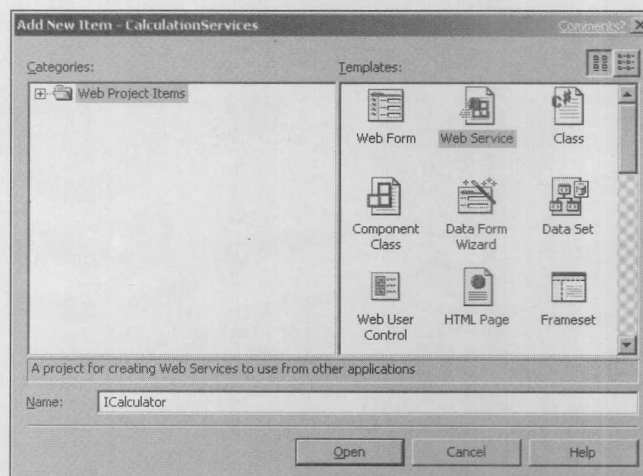
As you well know from traditional component-oriented programming, separating the interface from its implementation provides polymorphism between different implementations of the same service. The separation recognizes the fact that the basic unit of reuse in any application is the interface, not the object implementing it. You can apply this core principle of component-oriented programming to Web Service development as well.

In this case, the interface is a logical grouping of method signatures that act as the contract between the client and the Web Service provider. You can then switch between different providers with minimal or no changes to the client code because the client is written against an abstract service definition (the interface) rather than a particular service implementation.

The Web Services standard supports interfaces (referring to them as ports). But by default, Web Services support in .NET is method-based, not interface-based. So VS.NET, as it exists today, doesn't inherently allow you to develop interface-based Web Services. I'll show you the simple steps required—both on the server and cli-

ent—to compensate for the lack of interface-based Web Services in VS.NET, allowing you to develop and consume interface-based Web Services. But first I'll set up an example scenario.

Suppose you have a Web Service called Simple-Calculator that provides the four basic arithmetic operations—addition, subtraction, division, and



**Figure 1 Expose an Interface Definition.** To expose a Web Service interface definition, add a new Web Service item and give it an interface name. Clicking on the Open button causes VS.NET to create a skeletal Web Service. From there, remove all the implementation code from the wizard-generated code.

multiplication—and a client that consumes the Web Service. You implement the SimpleCalculator Web Service in .NET using C# (see Listing 1).

Simply add the [WebMethod] attribute to the methods you want to expose as Web Services—Add(), Subtract(), Divide(), and Multiply()—and .NET does the rest. Note that having WebService as a base class is optional; deriving from it gives you easy access to common ASP .NET objects, such as those for application and session states. You don't need these objects in this use-case. Also optional is the WebService attribute, but I strongly recommend you use it.

The WebService attribute lets you specify a Web namespace that contains your service, which you can use as you would a normal .NET namespace to reduce name collusion. If you don't specify a namespace, VS.NET uses <http://tempuri.org/> as a default. A published service usually uses a specific Uniform Resource Identifier (URI) as its namespace, typically the service provider company's name. The WebService attribute also allows you to provide a free-text description of your service that appears in the auto-generated browser page used by Web Service consumers and during development.

Writing the client code is almost equally trivial. Select Add Web Reference... from the client's project in VS.NET and point the wizard at the site containing the Web Service ASPX file. This makes VS.NET generate a wrapper class—called SimpleCalculator—that the client will use (see Listing 2). The SimpleCalculator wrapper class has all the methods the Web Service developer applied [WebMethod] to as public methods. The wrapper class (sometimes called a “Web Service Proxy” class) encapsulates completely the complex interaction with the remote object. The wrapper class is also the only entity coupled to the service's location—its base class, SoapHttpClientProtocol, has a property called Url that points to the object's location.

The client code uses the wrapper class as if the SimpleCalculator object were a local object:

```
SimpleCalculator calculator;
calculator = new SimpleCalculator();
int result = calculator.Add(2,3);
Debug.Assert(result == 5);
```

Clearly, VS.NET makes invoking Web methods trivial.

But you have a problem: The client ends up programming directly against the “object” providing the service (SimpleCalculator

## C# • It All Adds Up With the SimpleCalculator Class

```
[WebService(
    Namespace=
    "http://CalculationServices.com",
    Description = "The SimpleCalculator Web Service provides the four
    basic arithmetic operations for integers.")]
public class SimpleCalculator: WebService
{
    public SimpleCalculator(){}
    [WebMethod]
    public int Add(int num1,int num2)
    {
        return num1 + num2;
    }
    [WebMethod]
    public int Subtract(int num1,int num2)
    {
        return num1 - num2;
    }
    [WebMethod]
    public int Divide(int num1,int num2)
    {
        return num1 / num2;
    }
    [WebMethod]
    public int Multiply(int num1,int num2)
    {
        return num1 * num2;
    }
}
```

**Listing 1** To expose a class method as a Web Service, simply add the [WebMethod] attribute. Note that deriving from WebService is optional. Using the [WebService] attribute is optional also, but you should use it to provide a service description and containing namespace.

## C# • Encapsulate the Interaction

```
public class SimpleCalculator : SoapHttpClientProtocol
{
    public SimpleCalculator()
    {
        Url = "http://www.CalculationServices.com/SimpleCalculator.asmx";
    }

    [SoapDocumentMethod("http://CalculationServices.com/Add")]
    public int Add(int num1,int num2)
    {
        object[] results = Invoke("Add", new object[]{num1,num2});
        return (int)(results[0]);
    }

    // Other method wrappers
}
```

**Listing 2** The SimpleCalculator Web Service wrapper class—generated for the Web Service shown in Listing 1—encapsulates the interaction with the Web Service completely and shields the client from the details. It contains the service location in the public Url property, which is defined in the SoapHttpClientProtocol base class.

in this case) instead of against a generic abstraction of such a service. You want the SimpleCalculator Web Service polymorphic with a service abstraction—an interface.

For example, imagine the client wants to switch from the SimpleCalculator to a different calculator Web Service, called ScientificCalculator. ScientificCalculator supports the same interface as SimpleCalculator, but it's perhaps faster, cheaper, or more

accurate. You'd like to define a generic calculator interface, the ICalculator interface, and expose it as Web Service:

```
// Imaginary attribute.
// Does not exist in .NET
[WebService]
interface ICalculator
{
    int Add(int num1,int num2);
    int Subtract(int num1,int num2);
    int Divide(int num1,int num2);
    int Multiply(int num1,int num2);
}
```

Assuming you could do that (I'll show you how shortly), you can code against only the interface definition instead of a particular implementation of it (see Listing 3).

The only thing that changes in the client's code when it switches between service providers is the line that decides the exact interface implementation to use. You can even put that decision in a different assembly than the "main" client's logic, and you can only pass interfaces between the two. Another benefit you can reap from interface-based Web Services: The client can publish the interface definition, enabling different service vendors to implement the client's requirements more easily.

Now you're ready to write the server and client code to work around VS.NET's lack of support for interface-based Web Services.

## Define and Implement a Web Interface

To enable an interface-based Web Service, first expose the Web Service interface definition. For simplicity's sake, assume the service provider is responsible for both defining and implementing the interface (the client or any third party can expose the interface definition and have anybody implement it, but it requires additional steps).

Create a new Web Service project called CalculationServices. Right-click on the project and select Add Web Service... In the Add New Item dialog, type "ICalculator" as the interface name and click on Open (see Figure 1).

VS.NET then creates a skeletal Web Service called ICalculator. Open the ICalculator.asmx.cs file and change the ICalculator type definition from "class" to "interface." Remove the derivation from System.Web.Services.WebService, as well. An interface, by definition, has no implementation code—remove the constructor and the InitializeComponent() and Dispose() methods. Finally, remove the commented HelloWorld() method example.

Next, add the interface methods—Add(), Subtract(), Divide(), and Multiply(). Although in principle you could simply apply the [WebMethod] attribute to each interface method to expose the interface as a Web Service definition, you shouldn't in practice because of the [WebService] attribute. This attribute applies only to classes, and it's sealed—you can't subclass it or change it. So you can't assign a namespace and a description to the interface, which I advised you to do earlier. To clear this hurdle, you must provide an interface "shim"—an abstract class that exposes what looks like a pure interface

## C# • Prepare for a Service Provider Change

```
// Somewhere in the client code, it
// decides on the service provider:
ICalculator calculator = (ICalculator) new ScientificCalculator();

// or

ICalculator calculator = (ICalculator) new SimpleCalculator();

// This part of the client code is
// polymorphic with any provider of the service:

int result = calculator.Add(2,3);

Debug.Assert(result == 5);
```

**Listing 3** Ideally, you want your calculator Web Services polymorphic with a service abstraction—an interface. You can then switch between service providers with minimal or no changes to the client code.

**VS.NET, as it exists today, doesn't inherently allow you to develop interface-based Web Services.**

definition. In the ICalculator.asmx.cs file, add the ICalculatorShim pure abstract class definition:

```
[WebService(
    Name = "ICalculator",Namespace=
    "http://CalculationServices.com",
    Description = "This Web Service is only the definition of
    the interface. You cannot invoke method calls on it.")]
abstract class ICalculatorShim : ICalculator
{
    abstract public int
    Add(int num1,int num2);

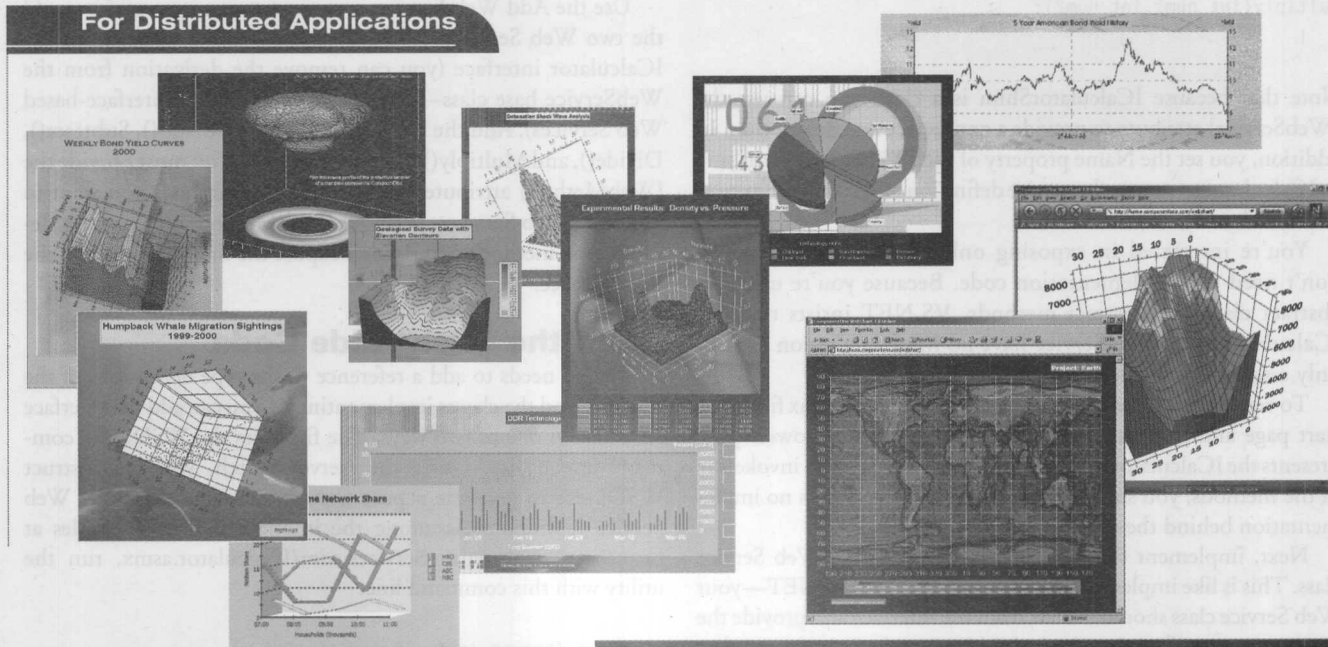
    abstract public int
    Subtract(int num1,int num2);

    abstract public int
    Divide(int num1,int num2);

    abstract public int
```



**For Distributed Applications**



**For Browser-independent Web Server Applications**

**Charts. Features. Performance.**

**The ultimate charting tools for distributed and Web applications.**

Need to create stunning, blazingly fast charts for your VC++®, Visual Basic®, Delphi™, and C++ Builder™ applications? Rely on the new versions of ComponentOne Chart™ 7.0 and ComponentOne WebChart™ 7.0! ComponentOne Chart 7.0 is the most powerful charting tool available to create high-performance 2D and 3D charts and graphs in minutes and distribute them royalty-free. To create browser-independent, dynamic charts for Active Server Page (ASP) applications, choose ComponentOne WebChart 7.0, with built-in ComponentOne Chart 7.0 technology. Easily develop any chart or graph and serve it as JPEG or PNG to any browser. You can also serve the charts as an ActiveX® control to allow your end-users to change chart types and data values for lightning-fast, interactive applications. **VC++ users can further harness the native power of the charts by calling the DLLs directly.** Features in both tools include: Area (now includes 3D effects), bar, bubble, candlestick, hi-lo-open-close, pie, plot, polar, radar, ribbon, scatter, stacking area, stacking bar, surface, 100% stacking, 3D scatter, 3D bar, and other chart types; Data binding to data sources, often without writing a single line of code; Intelligent update for fast, real-time charting and tracking; Built-in zooming, scaling, and rotation; Interactive interfaces, like drill-downs with "hit" detection; Annotations anywhere with unlimited labels; Dynamic date and time axes on any plot; Dual Y-axis and logarithmic axes; Scientific, accounting, currency, fraction, and percent annotations; Alarm zones; and more! Plus, ComponentOne Chart 7.0 offers you 2D and 3D Unicode controls for the best National Language Support available, and localization of DLL satellites so you can substitute your own error messages and property pages based upon locale.



**Trust the performance, support, and value you can only get from ComponentOne tools. Download your eval copies today!**

**TAKE ADVANTAGE OF OUR FABULOUS LIMITED TIME OFFERS:**

Competitive Upgrade for ComponentOne Chart 7.0.....	<b>\$359.<sup>95</sup></b>	Competitive Upgrade for ComponentOne WebChart 7.0.....	<b>\$659.<sup>95</sup></b>
ComponentOne Chart 7.0 Upgrade for Oletra Chart 6.0.....	<b>\$199.<sup>95</sup></b>	ComponentOne WebChart 7.0 Upgrade for Oletra Chart 6.0.....	<b>\$499.<sup>95</sup></b>
Full Version of ComponentOne Chart 7.0.....	<b>\$459.<sup>95</sup></b>	ComponentOne WebChart 7.0 Upgrade for True WebChart 6.0.....	<b>\$299.<sup>95</sup></b>
Premium Support with Subscription for ComponentOne Chart 7.0.....	<b>\$279.<sup>95</sup></b>	Full Version of ComponentOne WebChart 7.0.....	<b>\$759.<sup>95</sup></b>
		Premium Support with Subscription for ComponentOne WebChart 7.0.....	<b>\$459.<sup>95</sup></b>

**ComponentOne**

[www.componentone.com](http://www.componentone.com)

East Coast Office: 1.800.858.2739 or 1.412.681.4343 • West Coast Office: 1.888.228.4839 or 1.510.496.3240

ComponentOne™ LLC • 4516 Henry Street • Pittsburgh, PA 15213 • USA

© 2001 ComponentOne LLC. All rights reserved. All product names are owned by their respective holders.

```
Multiply(int num1,int num2);
}
```

Note that because `ICalculatorShim` is a class, you can use the `[WebService]` attribute to provide a namespace and description. In addition, you set the `Name` property of the `[WebService]` attribute to `ICalculator` to expose the service definition as `ICalculator` instead of `ICalculatorShim`.

You're interested in exposing only the signatures, so you don't need the implementation code. Because you're using an abstract class and abstract methods, VS.NET insists that the `ICalculatorShim` Web Service have no implementation code—only a service definition.

To verify that all is well so far, set the `ICalculator.asmx` file as the start page and run the project; the auto-generated browser page presents the `ICalculator` interface definition. If you try to invoke any of the methods, you should get an error because there's no implementation behind the service.

Next, implement the `ICalculator` interface on a Web Service class. This is like implementing any other interface in .NET—your Web Service class should inherit from the interface and provide the implementation for its methods. In this example, provide two class implementations: the `SimpleCalculator` and the `ScientificCalculator` Web Services (download Listing 4 from the *VSM* Web site; see the Go Online box for details).

Use the `Add Web Service...` context-menu item again to add the two Web Services. Add a derivation (inheritance) from the `ICalculator` interface (you can remove the derivation from the `WebService` base class—it bears no relevance to interface-based Web Services). Add the implementation to the `Add()`, `Subtract()`, `Divide()`, and `Multiply()` interface methods. You must provide the `[WebMethod]` attributes for the interface methods you want to expose as Web Services. Without `[WebMethod]` on a class implementation method, .NET won't expose the method as part of the Web Service.

## Write the Client-Side Code

The client needs to add a reference to the type definitions of the interface and the classes implementing it. You can add the interface reference in one of two ways. The first uses the `WSDL.exe` command-line utility. Using the `/server` switch, you can instruct `WSDL.exe` to generate a pure abstract class matching the Web Service definition. Assuming the interface definition resides at `http://www.CalculationServices.com/ICalculator.asmx`, run the utility with this command line:

```
WSDL.exe /server /out: ICalculatorDef.cs
http://www.CalculationServices.com/ICalculator.asmx
```

Then add the `ICalculatorDef.cs` source file to the client project. Unfortunately, even though .NET knows about interfaces, the `/server` switch generates a pure abstract class with abstract methods:

```
public abstract class ICalculator : WebService
{
    [WebMethod]
    [SoapDocumentMethodAttribute("http://
CalculationServices.com/Add")]
    public abstract int Add(int num1, int num2);
    // rest of the ICalculator methods
}
```

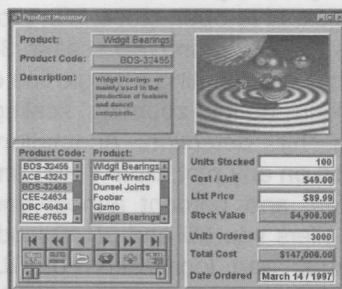
But what you need is an interface definition. Open the `ICalculatorDef.cs` file, remove the `WebService` base class, and change the `ICalculator` definition from abstract class to interface. Remove all the attributes (on `ICalculator` and its methods) and the public and abstract modifiers from all the methods. You should now have the original interface definition.

The second way a client can import the interface definition: Add a Web reference to the `ICalculator` Web Service, then extract the interface methods from the wrapper class. To do so, point the `Add Web Reference...` wizard to the site containing the interface definition. This generates a wrapper class called `ICalculator`, which exposes the original `ICalculator`'s methods as well as a Web Service wrapper class's other methods. You need only method definitions for an interface, so remove all the interface method bodies and the other methods completely, including the constructor. Remove the `SoapClientProtocol` base class and the "public" modifier on the interface method. Remove all class and method attributes. Lastly, change the `ICalculator` definition from "class" to "interface." Essentially, the client should now have the original interface definition.

Next, the client must add a Web reference to the Web Services that provide the interface's implementation. Again, using the `Add`

## The *Fastest* Database Engine

Discover **CodeBase**, the **fastest** database engine on the market that's also fully **xBASE** compatible. Query a million-record table in 0.34 seconds, or read 300,000 records in just 0.59 seconds. It's lightning quick!



### ADO/ODBC Support!

This application was created without a single line of source code! Use native and 3rd party controls with our new ADO/OLE-DB and ODBC support. Creating database programs has never been easier!

- ✓ Multi-user file compatible with FoxPro, Clipper and dBASE
- ✓ Client/Server, multi-user & single user support included
- ✓ Small DLL loads quickly, simply and use few resources
- ✓ 100% Portable: Windows, Pocket PC etc. 32 & 64 bit
- ✓ Supports VB, C, C++, Delphi, Java etc.
- ✓ Reader's Choice Award for 5 Years
- ✓ Royalty Free Distribution

**FREE 30 Day  
Test Drive!**

**SEQUITER**  
SOFTWARE INC.

www.sequiter.com - info@sequiter.com

Phone: (780) 437-2410  
Fax: (780) 436-2999

Web Reference... wizard, point the wizard to where those implementations reside. VS.NET generates wrapper classes for those implementations—SimpleCalculator and ScientificCalculator, in this example. These machine-generated wrapper classes won't mention ICalculator. To provide polymorphism with ICalculator, add a derivation from it. The SimpleCalculator and ScientificCalculator classes should now look like those in Listing 5, which you can download.

Here's the client-side design pattern: ICalculator provides the service definition. The Web Service's location is decided at the derived SimpleCalculator or ScientificCalculator classes. The wrapper classes know how to forward the calls to the Web Service, but not how it's implemented on the server side. In fact, all that distinguishes SimpleCalculator from ScientificCalculator is the encapsulated service location or provider.

Finally, you can now write interface-based, polymorphic Web Services code (see Listing 3). You can also make an interesting observation: From the client's perspective in the Web Services world, the location of the service—the URL—is the object.

Web Services will become part of almost every developer's career in the next few years. However, the supporting tools are immature compared to existing design methodologies and component-oriented technologies you've grown accustomed to in the intranet world. Today's challenge is how to combine the two. I hope this article convinces you not to give up on proven and elegant concepts just because VS.NET doesn't support them. With a bit of tweaking and the proper observation that a URL equals an object, you can employ established concepts in a brave new world. **VSM**

### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110WS** Download the code for this article separately. This article's code includes a basic, non-interface-based Web Service; two interface-based Web Services; a test client that uses them all; and Listings 4 and 5.

**VS0110WS\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

**Juval Lowy** is a seasoned software architect and the principal of IDesign, a consulting company focused on COM/.NET design. Juval also conducts training classes and gives conference talks on component-oriented design and development. He wrote the book *COM and .NET Component Services – Mastering COM+ (O'Reilly)*. Reach him through [www.componentware.net](http://www.componentware.net).



New  
Version

## PC-lint 8.0 for C/C++ Bug of the Month #651

```
#include <stdio.h>

struct { int a[3], b; } w[] = { { 1, 2, 3 }, 2 };

int main()
{
    printf( "w[0].b = %d\n", w[0].b );
    return 0;
}
```

The programmer expected to print the value '2'. Instead he got '0'. Why? Call if you need a hint or visit our web site at [www.gimpel.com](http://www.gimpel.com)

**PC-lint for C/C++** will catch this and many other bugs. It will analyze a mixed suite of C and C++ modules to uncover bugs, glitches, quirks and inconsistencies.

**Not your Grandpa's lint:** PC-lint has introduced several spectacular and revolutionary innovations in the art of static program analysis. Taking clues from initializers, assignments, and conditionals, variable and member values are tracked, enabling reports on potential uses of null pointers and out-of-bounds subscripts.

**New with Version 8:** Interfunction value tracking – Actual argument values are used to initialize parameters; return values are computed; a multi-pass operation (you control the number of passes) allows you to plumb the depths of function behavior to arbitrary levels.

**Plus Our Traditional C/C++ Warnings:** Uninitialized variables, inherited non-virtual destructors, strong type mismatches, ill-formed macros, inadvertent name-hiding, suspicious expressions, etc., etc.

**Full Language Support for ANSI/ISO C and C++.**

**PC-lint for C/C++ \$239**  
Numerous compilers/libraries supported.  
Runs on Windows, MS-DOS, and OS/2.

**FlexeLint for C/C++**  
The same great product for other operating systems. Runs on all UNIX systems, VMS, mainframes, etc. Distributed in shrouded C source form. Call for pricing.

**30 Day Money Back Guarantee**

## Gimpel Software

Serving the C/C++ Community for 15 Years.

3207 Hogarth Lane, Collegeville, PA 19426

**CALL TODAY (610) 584-4261 Or FAX (610) 584-4266**

[www.gimpel.com](http://www.gimpel.com)

PA add 6% sales tax.

PC-lint and FlexeLint are trademarks of Gimpel Software



# Build a Project Documenter Utility

Explore the VBDoc project to learn debugger tricks and shortcuts on your way to building a utility that documents your VB projects.

by Stan Schultes

## Technnology Toolbox

- ☐ VB.NET
- ☐ C#
- ☐ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☒ VB6

If I can say one thing about myself, it's that I'm lazy. I spend all kinds of time writing comments in my programs, but I hate spending time creating documentation describing the apps I've written. I do like to write code, however, so I decided to write a utility to help automate my documentation chores. This month's VBDoc utility extracts and formats comments in your VB6 projects to use as a starting point for your project documentation. Download the sample code from the *VSM* Web site and follow along (see the Go Online box for details).

Load and run VBDoc in the VB design environment to see how it works. The default project is VBDoc.vbp itself, or you can click on the Browse button to choose a different VB project. You can see a list of the project's files in the Files in Project listbox. Click on each filename in turn to see the header comments displayed in the Contents area. Header comments include the first contiguous block of comments in the file. If the file includes optional delimiters ('+= and '-= by default), the header comments include the delimiters and all lines in between. You set the desired delimiters with the Tools | Set Delimiters menu item. You can edit and save the header comments in the Contents area without modifying the source files.

Click on the Show Docs button to fill the contents area with formatted text. Because the Contents area is a RichTextBox control, VBDoc can use font styles to delimit sections in the documentation output. Project attributes are listed first, followed by the header com-

ments for each file in your project. You click on the Save Docs button to save the documentation to a Rich Text Format (RTF) file you can edit with Word or other applications (see Figure 1). The file defaults to the project name and directory, as in VBDoc.rtf.

Set breakpoints at important points in the VBDoc project so you can see how the code operates. Go to the Project Explorer (View | Project Explorer or Ctrl+R) in the VB design environment. Right-click on frmMain and choose View Code from the context menu, or click on the View Code icon at the top of the project window. Set a breakpoint on the first line in frmMain's CreateDocFile routine by clicking in the gray-colored left margin next to the line containing the GetPath call. VB highlights the line of code in red when you set a breakpoint. You also set breakpoints on the first executable lines in the ShowFileDoc and ShowDoc routines, as well as on the first executable lines in clsProject's ProjectFile Property Let procedure and in clsFile's Init method.

Start the VBDoc project by pressing the F8 key. F8, the Step-Into function key, starts the program on the first executable line in the startup form's Form\_Load event. The yellow highlight represents the next statement to execute. Continue pressing F8, and on the third repetition the debugger steps into the InitSettings routine. Press F8 a couple more times, then use the Step-Out function key (Ctrl+Shift+F8) to continue running the program until you're back at the line in Form\_Load following the call to InitSettings.

Now use the Step-Over function key (Shift+F8)

to continue through the Form\_Load routine without seeing each lower-level routine execute. When you step over the line in Form\_Load that sets the default project path, VB stops at your breakpoint in the clsProject's ProjectFile Property Let routine:

```
If InStr(UCase$(ProjectFilename), _
    ".VBP") Then
```

You can see how VB arrived at this point by choosing the View | Call Stack menu item (or by pressing Ctrl+L). You can jump into the code at any point in the call stack by double-clicking on a routine name in the list. The line labeled Non-Basic Code indicates that VB fired an event routine on your behalf. Continue from your breakpoint by pressing the Run key (F5), and the debugger then stops at the breakpoint in clsFile's Init method for the first file. For now, continue by pressing the Run key (F5) repeatedly—once for each file in the project—until the main form shows.

VBDoc's key element is a pair of class modules, clsProject and clsFile. Restart your project and run it (F5) to the first breakpoint in clsProject's ProjectFile Property Let procedure. Step over each line (Shift+F8) until you reach the LoadFile call. Step into LoadFile (F8) to see how the contents of the VBP file are returned:

```
Public Function LoadFile _
    (ByVal FileName As String) As String
    Dim iFileNum As Integer
    iFileNum = FreeFile
    Open FileName For Input As #iFileNum
    LoadFile = Input(LOF(iFileNum), #iFileNum)
    Close #iFileNum
End Function
```

The Input statement reads the number of bytes specified by the Length of File (LOF) function. Use Step-Out (Ctrl+Shift+F8) to return to the ProjectFile routine.

Next, step into the ParseProject function to load the project's properties and files (see Listing 1). The process of reading a file and taking apart its contents is called *parsing*. ParseProject first uses the Split function to separate the VBP file by carriage return-line feed characters using the vbCrLf constant. This is a quick way of breaking the file into individual lines. Individual properties in the VBP file are composed of key-value pairs:

```
Class=clsProject; CProject.cls
```

You loop and split each line in the file using the = character to separate each key name from its value, and you pick up individual project properties and filenames with a Case statement. You create an instance of clsFile in ParseProject for each file in the project, and you call the Init property to load and analyze the file:

```
Case "FORM", "MODULE", "CLASS"
    'project files
    If colFiles Is Nothing Then _
        Set colFiles = New Collection
    Set oFile = New clsFile
    oFile.Init sFileType, sFileName, _
        GetPath(ProjectFile)
```

```
colFiles.Add oFile, oFile.FileName
Set oFile = Nothing
```

You store each file instance (one for each file in the project) in the colFiles collection using the filenames as collection keys.

Continue stepping the debugger or use the Run key (F5) until you come to the breakpoint in the clsFile Init routine, where you collect file attributes and read the file contents using the LoadFile routine:

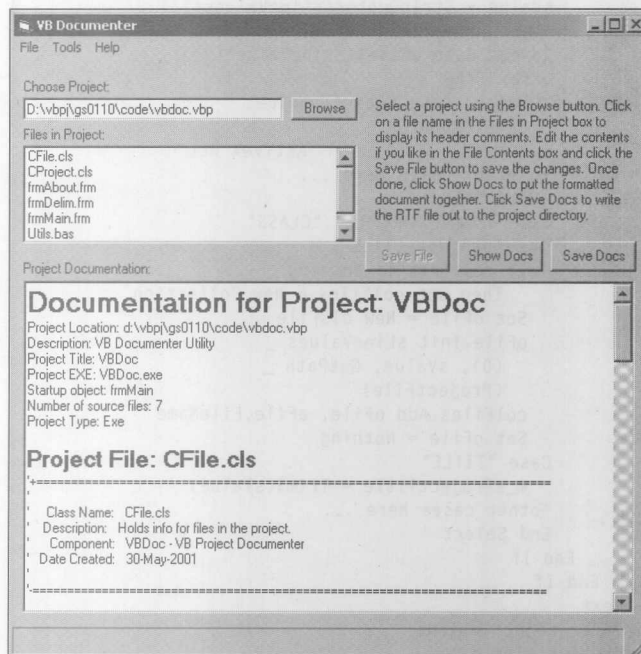
```
m_sFileContents = LoadFile _
    (GetFilePath(BasePath, m_sFileName))
```

The GetFilePath function determines the physical path to your project files. You need this function if some of your project's source files come from outside your source directory. This is common in situations where you have a library of common modules, classes, and forms in a common directory that you reuse on projects. If your source files all come from the same directory (as with VBDoc), the GetFilePath function returns the project directory. Continue pressing F5 to run at each breakpoint until the main form shows again.

## Parse Project Files

Now let's look at how you collect file header comments. Click on a filename in the Files in Project list, and VB stops at your breakpoint in frmMain's ShowFileDoc routine:

```
Set oFile = goProject.FileRef(FileName)
rtbProjectDoc.Text = oFile.FileHeader
```



**Figure 1 Off to a Good Start.** The VBDoc utility loads properties and a list of included modules from a VBP (VB Project) file. You extract the header comments from each module and edit them in the VBDoc form, then save the results to an RTF file that you can load in your favorite word processor. VBDoc doesn't finish the job of documenting your code, but it can start it.

The `goProject.FileRef` property returns a file object reference from the project's Files collection. Step into the `oFile.FileHeader` routine to see how VBDoc handles file header analysis:

```
Public Property Get FileHeader() As String
'returns the File Header comments
    If Len(m_sFileHeader) = 0 Then
        m_sFileHeader = ParseHeader()
        GetLineCount
    End If
    FileHeader = m_sFileHeader
End Property
```

#### VB6 • Parse the VB Project File

```
Private Function ParseProject _
    (ByVal FileContents As String) As Boolean
'returns comma-delim list of project
'files; empty string if invalid file
Dim sLineBuffer() As String
Dim sLineValues() As String
Dim sValue As String
Dim lLine As Long
Dim oFile As clsFile
'split vbp into individual lines
sLineBuffer = Split _
    (FileContents, vbCrLf)
For lLine = 0 To UBound(sLineBuffer)-1
    If Len(Trim$(sLineBuffer _
        (lLine))) Then
        sLineValues = Split _
            (sLineBuffer(lLine), "=")
        If UBound(sLineValues) > 0 Then
            sValue = StripQuotes(sLineValues(1))
            'save properties for keywords
            Select Case UCase$(sLineValues(0))
            Case "TYPE"
                Select Case UCase$(sValue)
                Case "OLEDLL"
                    m_sProjectType = "ActiveX DLL"
                    'other cases here ...
                End Select
            Case "FORM", "MODULE", "CLASS"
                'project files
                If colFiles Is Nothing _
                    Then Set colFiles = New Collection
                Set oFile = New clsFile
                oFile.Init sLineValues _
                    (0), sValue, GetPath _
                    (ProjectFile)
                colFiles.Add oFile, oFile.FileName
                Set oFile = Nothing
            Case "TITLE"
                m_sProjectTitle = Trim$(sValue)
                'other cases here ...
            End Select
        End If
    End If
Next
ParseProject = True
End Function
```

**Listing 1** The `ParseProject` function decomposes a VBP file into a set of properties and a collection of files. First it splits the file into individual lines, then it separates the key-value pairs to store each property value. You handle files by creating a file object and calling its `Init` method to get the file properties and contents.

If the file header for this file hasn't been processed already, you call the `ParseHeader` routine to extract the comments (download Listing A). `ParseHeader` separates the file into individual lines and searches for the header comment block. Header comments are found either between the optional delimiters or as the first contiguous block of comments.

Finally, let's see how you create formatted output. Click on the Show Docs button, and VB stops at your breakpoint in `frmMain's ShowDoc` routine (download Listing B). You format text programmatically in the `RichTextBox` by selecting the text and applying formatting to the selected text. You use the `sFormat` array in `ShowDoc` to keep track of where formatting is required in the output. The `sFormat` array contains a series of comma-delimited entries that specify the beginning character position in the text, the offset, and a descriptive name for the format ("Heading1", "Heading2").

Step through the code in `ShowDoc` to see how VBDoc creates the output from the project and file properties. Click on this line near the bottom of `ShowDoc`:

```
FormatDoc sFormat
```

Use the Run to Cursor keyboard shortcut (Ctrl+F8) to run to the selected line. Step into the `FormatDoc` routine to see how the `RichTextBox` formatting is applied (download Listing C). Here, you loop through the `sFormat` array, select the appropriate text, and apply the appropriate font style for the format name.

Continue running the app, and click on the Save Docs button to run to your final breakpoint in `frmMain's CreateDocFile` routine. You create the output file using the `SaveFile` method of the `RichTextBox`:

```
rtbProjectDoc.SaveFile sFile, rtfRTF
```

And that's it! Try enhancing VBDoc on your own by listing referenced objects (such as OCX files) or all the routine declarations in each file. Use the debugger shortcuts you've learned to debug your new features. Then rest easy knowing your project documentation chores have just gotten easier. **VSM**

**Stan Schultes** is a project and IT manager and a VB/Web enterprise application architect and developer in Sarasota, Fla. Stan is an MCP in VB and spoke on VB development at Microsoft's DevDays conference. He is a contributing editor for *Visual Studio Magazine* and writes regularly for it. Reach Stan at [stan@vbexpert.com](mailto:stan@vbexpert.com).

#### Go Online

Use these Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of VSM.

**VS0110GS** Download the code for this article separately. The code for this article includes a complete utility to create VB project documentation, as well as Listings A, B, and C.

**VS0110GS\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).





Based on TPC-C benchmark results, DB2 results: 440,879.95 tpmC, \$32.28/tpmC, availability date: 12/7/00. Oracle results: 220,667.27 tpmC, \$43.30/tpmC, availability date: 4/13/01. Results as of 4/11/01. TPC, TPC-C and tpmC are trademarks of the Transaction Processing Performance Council (www.tpc.org). IBM, the e-business logo and DB2 are registered trademarks of International Business Machines Corporation in the United States and/or other countries. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other company, product and service names may be trademarks or service marks of others. © 2001 IBM Corporation. All rights reserved.

THE VISITORS' UNIVERSE NEEDED CUTTING-EDGE SOFTWARE. THIS UNIVERSE HAD IT.

# DB2<sup>®</sup> FOR WINDOWS<sup>®</sup> 2000

■ TWICE THE POWER OF ORACLE • A FRACTION OF THE PRICE ■

@ business software [ibm.com/db2/windows](http://ibm.com/db2/windows)

IT'S A DIFFERENT KIND OF WORLD.  
YOU NEED A DIFFERENT KIND OF SOFTWARE.

# Don't Overuse Inheritance

Animal classification provides a good pattern for inheritance.

by Bill Wagner

## Technology Toolbox

- ☐ VB.NET
- ☐ C#
- ☐ SQL Server 2000
- ☐ ASP.NET
- ☐ XML
- ☐ VB6
- ☒ Other:

Some object-oriented experience and a desire to learn more

You've probably seen examples of object-oriented programming that use animal classification to explain inheritance. If done correctly, this analogy works well. In this column, I'll use animal classification to discuss what inheritance means, when to use it, and when other alternatives support your particular design better.

One way of understanding inheritance is to use the phrase "is a specialization of." For example, "a bear is a specialization of a mammal." Biologists classify animals into kingdoms, phyla, classes, orders, families, genera, and species (see Figure 1). Software developers can learn a lot from this classification. It's based on a hierarchy that moves downward from the general (kingdom) to the specific (species).

Two terms, *substitutability* and *coupling*, can help you understand the good and the bad of inheritance. Substitutability means you can use a derived class (bear) as a substitute anytime you expect the base class (mammal). In code, this means any method that

expects a reference to a mammal works correctly with a reference to a bear.

For example, suppose you're watching bats and flying squirrels. You might decide mammals need a `Fly()` method. Well, bears don't fly. This problem brings up coupling, which defines how closely two classes depend on each other. In the preceding example, the bear class and the mammal class are highly coupled. In fact, an inheritance relationship creates the highest possible coupling between two classes. The derived class and the base class are closely coupled. Any change in the base class affects the derived class, and any changes in implementing the base class affect the derived classes.

In general, you should minimize both coupling and inheritance. However, inheritance provides greater reuse than any other technique. A bear class supports all the methods in the mammal class automatically. A bear also inherits the implementation of all the methods in the mammal class. (See the sidebar, "Follow

## Follow These Class-Building Rules

I like to use real-world analogies to describe object-oriented techniques. For example, children playing with Legos can build many different objects. The key to building so many different things with Legos is that each piece is small and simple. Some new Lego sets contain larger and more complicated pieces. These more complex pieces are interesting, but more limiting. Keep this in mind when you design objects: The simpler the class or interface, the more ways you can use it. When you're designing and building classes, try to keep as many classes as simple as possible. Follow these rules:

- Ensure class inheritance follows the "is a" relationship. Or, use "is a specialization of" if that's easier.
- Look out for "is almost a," which usually implies the need for a new class to separate the nonimplemented functionality.
- Look out for "works like." This should be an interface or a contained object; the one you choose depends on the public requirements for the new interface.
- Make interfaces for reusable functionality.
- If you can get the behavior you want using containment or interfaces, use that instead of inheritance.
- Build more complicated objects and classes only when you need them in larger projects. Each class should add only one piece of clearly defined functionality.

## Resources

*Design Patterns: Elements of Reusable Object-Oriented Software*  
by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides  
[Addison-Wesley, 1995, ISBN: 0201633612]



These Class-Building Rules,” for more class-building information.)

A second reason for using inheritance is to provide a common interface that several classes support. All mammals support a common set of behaviors, or interfaces. However, you can add behavior unrelated to a class’s main purpose by defining interfaces a class can support. Good examples of interfaces are .NET’s `ISerializable` and `IDisposable` interfaces. These interfaces are unrelated to where a class fits in a hierarchy. Interfaces often live in the solution domain. They don’t represent real-world objects or abstractions. Rather, interfaces often represent software abstractions that add functionality to your real-world abstractions. An `IFly` interface is a good way to incorporate flying capability into the animal kingdom. Bats fly. Some birds fly. Some insects fly. These particular classes can support the `IFly` interface without tying them together using inheritance.

## Understand Common Problems

In real-world design, one mistake comes from trying to work with a hierarchy where a base class is too specific. For example, imagine you wrote the mammal base class after only seeing canines of various forms (dogs, wolves, coyotes, foxes). The first time you see a cat, you know it’s related to dogs, but your mammal class has methods that make no sense for a cat, such as running in packs, barking, and growling.

Your first instinct is to make a cat behave like a dog. The reason is obvious—you can reuse lots of code. But this inheritance will cause you problems. When you see yourself making a new derived class that doesn’t support all the base class methods naturally, find a different design. The right answer is to break the dog class into two different parts: the parts common to cats and dogs, and the parts unique to dogs. Then, both cats and dogs derive from this new base class. Coding this kind of mistake causes you to downcast objects to see the actual runtime type of an object before you use it:

```
if (pAnimal is (dog))
    pAnimal->bark();
```

Unfortunately, once you write this code, it’s too late. Look for these kinds of problems before you start coding.

Another mistake is using an interface as

a base class. You’ll run into problems because you’ve constrained a particular behavior to one group of classes. As a result, you limit the clients’ ability to design their own hierarchies. For example, consider flying animals. Flying is a good candidate for an interface. Many unrelated animals fly, including bugs, many birds, and bats. You might be tempted to create a `Fly()` method in the bird base class, but if you did, you’d commit the first mistake: Neither ostriches nor emus fly. Once again, you want to treat flying animals the same way, so you should make flying an interface. Bats, most birds, and many bugs can all implement the `IFly` interface.

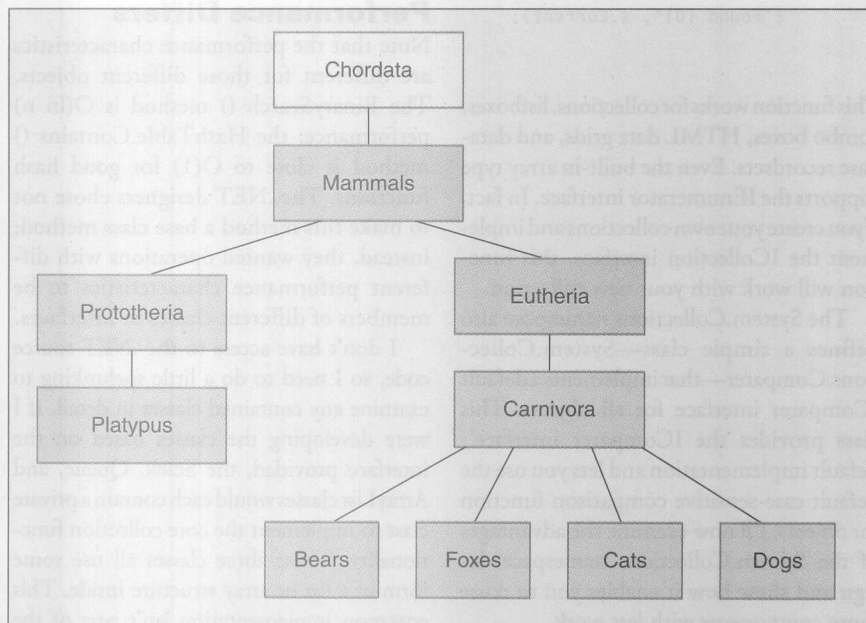
Maybe you’re thinking interfaces let you reuse only an interface description, but they don’t let you reuse implementation. This isn’t true—you can reuse the implementation by creating a class known as a *delegate*. This term is more generic than a C# delegate, so don’t confuse the two. A delegate class provides a common implementation that many different, unrelated classes can share. Consider the bird class again. Most birds fly, so it makes sense to create a class that can implement that ability. Then, any class derived from bird can contain a member of this `FlyingBird` class to reuse the flying capability. This move lets you reuse the implementation for flying birds without

cluttering the bird class’s public interface with methods not supported by all birds.

Now look at some real classes and see how Microsoft .NET uses these concepts to create an efficient, reusable design in the base class libraries. The `System.Collections` namespace contains collection classes and interfaces to work with those collections. Different storage shapes and performance characteristics distinguish the classes. The interfaces define common operations supported by different collection classes. The `System.Collections` namespace also includes a class that provides default implementations of the most common interfaces.

`ArrayList`, `BitArray`, `HashTable`, `NameValueCollection`, `ObjectList`, `Queue`, `SortedList`, `Stack`, and `StringCollection` are all collections of different types. These collections implement lists, arrays, maps, and hash table collections. These collections all inherit from the `System.Object` class and don’t share a common collection base class. The interfaces in the `System.Collections` namespace define the shared functionality in the collections.

The `System.Collections` namespace contains several interfaces, including `ICollection`, `IComparer`, `IDictionary`, `IDictionaryEnumerator`, `IEnumerator`, `IEnumerator`, `IHashCodeProvider`, and `IList`. Each collection implements the interfaces meaningful to it.



**Figure 1 Classify the Animals.** Notice in these object types that the common traits are always true, not “almost always true.” Behavior not shared by all groups is moved down the hierarchy, which contains as many layers as necessary to make sure all members share common elements.



For instance, the `HashTable` and `SortedList` classes are the only collections implementing the `IDictionary` interface, which defines methods and properties for retrieving key/value pairs (see which classes implement which interfaces in Table 1).

Looking at the `System.Collections` namespace in isolation, you might think it would be simpler to use base classes instead of interfaces. The true power of this design comes out when you look outside the `System.Collections` namespace. Many classes outside the `System.Collections` namespace implement the `ICollection` interface. The `ActiveX Data Objects .NET` (ADO .NET) `DataSetView` and `DataGridView` classes implement `ICollection`. The `WinForms` `ComboBox` and `ListBox` classes provide `ICollection` interfaces to data. The `WebForms` `DataGrid` class provides an `ICollection` interface to examine the data in a grid. By using techniques other than simple inheritance, the .NET designers provide much greater reuse between objects storing groups of items. Using the `IEnumerable` interface, `ICollection`'s base interface, you can write a simple function that prints items from a collection to the console:

```
Void copy_items (IEnumerable Src){
    IEnumerator s =
        Src.GetEnumerator ();
    While (s.MoveNext ())
        System.Console.WriteLine
            ("Found {0}", s.Current);
}
```

This function works for collections, listboxes, combo boxes, HTML data grids, and database recordsets. Even the built-in array type supports the `IEnumerator` interface. In fact, if you create your own collections and implement the `ICollection` interface, this function will work with your new collection.

The `System.Collections` namespace also defines a simple class—`System.Collections.Comparer`—that implements a default `IComparer` interface for all objects. This class provides the `IComparer` interface's default implementation and lets you use the default case-sensitive comparison function for objects. I'll now examine the advantages of the `System.Collections` namespace design and show how it enables you to reuse more components with less work.

First, consider why collections have no base class. You can't say anything about the collections that is always true about all collections and only the collection classes. Try

Classes	Implemented Interfaces
<code>ArrayList</code>	<code>ICollection</code> , <code>IEnumerator</code>
<code>BitArray</code>	<code>ICollection</code> , <code>IEnumerator</code> , <code>ICloneable</code>
<code>CasInsensitiveComparer</code>	<code>IComparer</code>
<code>CasInsensitiveHashCodeProvider</code>	<code>IHashCodeProvider</code>
<code>Comparer</code>	<code>IComparer</code>
<code>HashTable</code>	<code>IDictionary</code> , <code>ICollection</code> , <code>IEnumerator</code> , <code>ISerializable</code> , <code>IDeserializationEventListener</code> , <code>ICloneable</code>
<code>Queue</code>	<code>ICollection</code> , <code>IEnumerator</code> , <code>ICloneable</code>
<code>SortedList</code>	<code>IDictionary</code> , <code>ICollection</code> , <code>IEnumerator</code> , <code>ICloneable</code>
<code>Stack</code>	<code>ICollection</code> , <code>IEnumerator</code> , <code>ICloneable</code>
<code>StringCollection</code>	<code>ICollection</code> , <code>IEnumerator</code> , <code>ICloneable</code>

**Table 1 Understand System.Collections Classes and Interfaces.** The `System.Collections` namespace provides a great case study in object-oriented design and inheritance. The collections don't share a common base class; rather, they share sets of interfaces you can implement by collections or other collection-like objects. You can decide to implement interfaces on a case-by-case basis.

a few statements: "All collections, and only collections, can store multiple objects." This statement is false because many other objects store multiple objects, including listboxes, XML DOM objects, data grids, and database tables. "All collections provide ways to search for objects" is also false. `HashTables` can find a key quickly. `ArrayList` provides a `BinarySearch` method. Database tables find keys.

## Performance Differs

Note that the performance characteristics are different for those different objects. The `BinarySearch()` method is  $O(\ln n)$  performance; the `HashTable.Contains()` method is close to  $O(1)$  for good hash functions. The .NET designers chose not to make this method a base class method; instead, they wanted operations with different performance characteristics to be members of different classes or interfaces.

I don't have access to the .NET source code, so I need to do a little spelunking to examine any contained classes in detail. If I were developing the classes based on the interface provided, the `Stack`, `Queue`, and `ArrayList` classes would each contain a private class to implement the core collection functionality. These three classes all use some form of a list or array structure inside. This common implementation isn't part of the interface, so it can be hidden safely from the user. All these classes might use the built-in array type or a collection of arrays to implement the storage for the collections. If you

look at `mscorlib.dll` with the Microsoft Intermediate Language (MSIL) Disassembler, you can examine the collection classes' contents. `Queue`, `ArrayList`, and `Stack` all use a built-in array of `System.Object` objects in their implementations.

Inheritance is a powerful tool and a key building block to any object-oriented language. Use it at the right time, and you can build a powerful set of classes you can reuse in many places. Use it at the wrong time, and you'll introduce special cases where code doesn't function the way you want. **VSM**

**Bill Wagner** is a founder of SRT Solutions ([www.srtsolutions.com](http://www.srtsolutions.com)). He has 15 years of programming experience, has written for *Visual C++ Developers Journal* and *VSM* on C++ and C# topics, and is the author of the upcoming *C# Core Language Little Black Book* (The Coriolis Group). E-mail Bill at [wwagner@srtsolutions.com](mailto:wwagner@srtsolutions.com).

## Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110BP\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

# SECURELY LICENSE AND DISTRIBUTE SOFTWARE



**SentinelLM™**  
Software-based solution  
for Electronic License Management

**SentinelExpress™**  
Software-based solution  
for Electronic Software Distribution

## Benefits

- Secures applications from unauthorized execution
- Protects developers and ISV's from illegal licensing and distribution
- Enables eCommerce for electronic software distribution

## Don't Let Others Compromise the Integrity of Your Software

SuperPro	SuperPro 797	SuperPro USB	Benefits
			<ul style="list-style-type: none"> <li>■ Offers complete software protection that is easily integrated and simple to deploy</li> <li>■ Secures applications with powerful multiple-algorithm protection and 128-byte memory</li> <li>■ Enables remote software updating and demo upgrading to fully-licensed versions</li> </ul>

For more information please visit: [sentinel.rainbow.com](http://sentinel.rainbow.com)

50 Technology Drive, Irvine, CA 92618  
1.800.852.8569 • [www.rainbow.com](http://www.rainbow.com)  
[sales@rainbow.com](mailto:sales@rainbow.com)

**RAINBOW**  
TECHNOLOGIES

©2001 Rainbow Technologies, Inc.

# Whip Forms Into Shape

Use VB.NET's runtime-defined event handlers and property extenders to get consistent event behavior in your forms.

by Kathleen Dollard

## Technology Toolbox

### ☒ VB.NET

### ☐ C#

### ☐ SQL Server 2000

### ☐ ASP.NET

### ☐ XML

### ☐ VB6

### ☒ Note:

Requires VB.NET beta 2 or later.

## Resources

"Get Started With WinForms" by Ken Getz and Paul D. Sheriff  
[Visual Basic Programmer's Journal  
April 2001]

A lot of boring code to provide consistent validation probably winds up in your apps. If I had ruby slippers, I'd click my heels three times and wish for a simple tool to do this job for all our apps. With a tool to provide items such as `IsRequired` and `IsNumeric` in the Properties window, you'd have to do nothing more than set these properties to have consistent event behavior in all your forms.

No ruby slippers necessary: It's surprisingly easy to do this in Visual Basic .NET (VB.NET). You can link together two simple new concepts—runtime-defined event handlers and property extenders—to provide powerful functionality.

An event is an occurrence, such as a user action, that causes a message to be sent to your application. You define an event procedure (also called an event handler) in your code to respond to this message. VB6 defines event procedures by matching the event procedure's name with the name of the event. For example, `Button1_Click` handles the `Button1` control's Click event.

VB.NET allows greater flexibility in hooking events and naming event procedures. It has at least three ways to hook up events: the VB6 compatibility class's pseudo control array classes, the `Handles` clause of event procedures, and the `AddHandler` function. Each approach is more flexible than the previous.

The compatibility class's control array classes work for converted code, but they aren't easy to use in new code or in adding controls to existing control arrays. You generate a `Handlers` clause automatically if you double-click on a control in VB.NET:

```
Private Sub Button1_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles _
    Button1.Click
```

The default name is familiar, but you can use any name you want. The sender is the object that raised the event—in this case, the `Button1` control. The second parameter is an instance of the `System.EventArgs` class. In VB.NET, the second argument is either an instance of the `System.EventArgs` class or one of the classes that inherits from it.

## Provide an Event Procedure

`Windows.Forms` declares many events in the `Control` class, which means these events are available for all `Windows.Forms` controls that inherit from the `Control` class. Many of these events use the `System.EventArgs` class, although some such as `KeyPress` have a specific event argument class. You can view the control hierarchy in the object browser.



The Handles clause can specify one procedure to handle multiple control events. It's the easiest way to provide a single event procedure for a set of controls. The drawbacks: It's difficult to ensure that all controls have a consistent set of events, and it's a design-time tool not available at run time.

To ensure a consistent set of event procedures, setting them in code at run time is more reliable even if it's a bit more work. Use the AddHandler function to hook events to event procedures at run time:

```
Dim Required() As Control = {txtLastName, txtDOB}
Dim ctl As Control
For each ctl In Required
    AddHandler ctl.Leave, _
        AddressOf Required_Leave
    AddHandler ctl.Change, _
        AddressOf Required_Change
    AddHandler ctl.Change, _
        AddressOf Standard_Change
Next
```

This code adds three event procedures to all controls in the array. Two event procedures are raised when the ctl.Change event occurs. This works fine, with the only caveat being not to rely on the order in which the events are fired.

The extender component model enables you to add specific extender properties to all, or a subset of, the controls on a form. These pseudo properties appear in each control's Properties window (see Figure 1). However, the data is contained in the extender, not in the control itself.

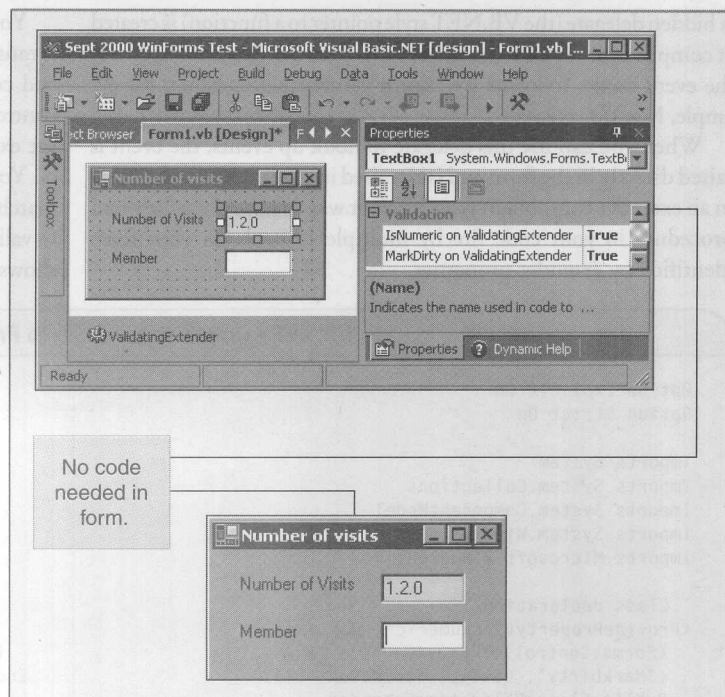
The extender class's declaration includes attributes that indicate it's an extender and what properties it supports:

```
<ProvideProperty("IsRequired", GetType(Control)), _
    ProvideProperty("MarkDirty", GetType(Control))> _
Public Class ValidatingExtender
    Inherits Component
    Implements IExtenderProvider
```

This extender (like most extenders) inherits from the System.ComponentModel.Component, so you can add it to the toolbox and drop it onto a form, where it resides in the component window. Extenders must implement the IExtenderProvider interface, which contains a single method, CanExtend. The CanExtend method determines whether the extender properties are included for a particular control.

Several things must happen for the extender properties to appear in a control's Properties dialog. In the component, you must declare the class with an attribute specifying the property, CanExtend must return True, and there must be a Public SetExtenderPropertyName and/or a GetExtenderPropertyName (see Listing 1). In the project where you'll use it, you must reference the component DLL, place the component in the toolbox through Customize Toolbox (right-click on the toolbox), and drag an instance of the control onto your form. .NET beta 2 has a glitch where the extender sometimes behaves badly if you don't place the extender DLL in the solution folder or register it in the global assembly cache.

VB persists these extender properties automatically with the form.



**Figure 1 Add Extender Properties.** Turn on automatic and consistent validation by simply dropping a component on your form and setting properties. Create an extender that adds new properties to the Properties dialog for controls on your form, and add consistent event handling for validation or other purposes to your control when you set these properties.

You need additional attributes only for special persistence handling. Similarly, the extender properties appear in the Properties window by default, but you can suppress them through additional attributes.

You access extender properties differently through code because they belong to the extender, not the control. You might want to set these properties at run time, perhaps using the database field types to determine appropriate validation. This is perfectly valid, but you must use this syntax:

```
ValidatingExtender.SetIsNumeric(txtVisits, True)
```

Unfortunately, an IDE bug in beta 2 results in an IDE crash if you paste a control onto a form using an extender. This problem occurs with your extenders, the VB6Compatibility library control array extenders, and the tooltip extender.

## Hook Up Events

The component in Listing 1 does the real work. It's responsible for hooking up the events and the event procedures. It also keeps track of the objects and their extender properties using a hash table (collection).

SetIsRequired and SetMarkDirty are called automatically when the class is created at both design time and run time. These methods contain logic to determine whether the event procedure should be hooked to the event, and they call AddHandler to add the hook if needed. Checking the DesignMode status keeps the event procedures from being added at design time.

The SetMarkDirty procedure demonstrates an undocumented feature of events raised in classes, which Bill McCarthy showed me.

A hidden delegate (the VB.NET style pointer to a function) is created at compile time for each declared event. This delegate is named with the event name, followed by "Event" (<eventname>Event; for example, MarkDirtyEvent).

When you can use this delegate to hook up events, the event is raised directly in the form and isn't raised in the extender. This trick in an extender component is an efficient way to have the same event procedure in your code fire by multiple controls on your form identified by extender properties.

You can hook up as many events to the control as you like. This means you can have both default handling specified in your extender and control-specific handling specified in the form for the same control event. You can also add normal methods and properties to the extender component, just as you would any other class.

You can customize the component class shown in Listing 1 to match your own style of control validation. Numerous approaches to validating data rely on specific events being fired, and this listing shows only how to ignore non-numeric keystrokes.

### VB.NET • Create an Extender to Provide Event Handling

```
Option Explicit On
Option Strict On

Imports System
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows
Imports Microsoft.VisualBasic

' Class declaration
<ProvideProperty("IsNumeric", GetType _
  (Forms.Control)), ProvideProperty _
  ("MarkDirty", GetType(Forms.Control))> _
Public Class ValidatingExtender

  Inherits ComponentModel.Component
  Implements IExtenderProvider

  Public Event MarkDirty As EventHandler

  Private mIsNum As New Hashtable()
  Private mMark As New Hashtable()

  #Region "Component Designer generated code"

  Public Function CanExtend( _
    ByVal extender As Object) As Boolean _
    Implements IExtenderProvider.CanExtend
    Return (TypeOf extender Is Forms.Control) _
      And (Not TypeOf extender Is Forms.Label) _
      And (Not TypeOf extender Is Forms.Form) _
      And (Not TypeOf extender Is Forms.Button)
  End Function

  <ExtenderProvidedProperty(), Description _
    ("Numeric Entry?", Category "Validation")> _
  Public Function GetIsNumeric( _
    ByVal ctl As Forms.Control) As Boolean
    Return CBool(mIsNum(ctl))
  End Function

  <ExtenderProvidedProperty()> Public Sub _
    SetIsNumeric(ByVal ctl As Forms.Control, _
    ByVal IsNum As Boolean)
    Dim IsSet As Boolean
    If Not ctl Is Nothing Then
      If (Not mIsNum(ctl) Is Nothing) Then
        IsSet = CBool(mIsNum(ctl))
      End If
      mIsNum(ctl) = IsNum
    End If

    If Not DesignMode Then
      If IsNum And Not IsSet Then
        AddHandler ctl.KeyPress, _
          AddressOf Me.Num_KeyPress
        'More handlers as needed
      ElseIf Not IsNum And IsSet Then
        RemoveHandler ctl.KeyPress, _
          AddressOf Me.Num_KeyPress
        'Remove additional handlers
      End If
    End If
  End Sub

  <ExtenderProvidedProperty(), Description _
    ("Dirty event raised?", Category _
    ("Validation"))> Public Function GetMarkDirty _
    (ByVal ctl As Forms.Control) As Boolean
    Return CBool(mMark(ctl))
  End Function

  <ExtenderProvidedProperty()> Public Sub _
    SetMarkDirty(ByVal ctl As Forms.Control, _
    ByVal bMark As Boolean)
    Dim IsSet As Boolean
    If Not ctl Is Nothing Then
      If (Not mMark(ctl) Is Nothing) Then
        IsSet = CBool(mMark(ctl))
      End If
      mMark(ctl) = bMark
    End If
    If Not DesignMode Then
      If bMark And Not IsSet Then
        AddHandler ctl.TextChanged, _
          Me.MarkDirtyEvent
      ElseIf Not bMark And IsSet Then
        RemoveHandler ctl.TextChanged, _
          Me.MarkDirtyEvent
      End If
    End If
  End Sub

  Private Sub Num_KeyPress(ByVal sender As _
    Object, ByVal e As Forms.KeyPressEventArgs)
    'Sample of event proc.
    e.Handled = ((e.KeyChar < "0") Or _
      (e.KeyChar > "9")) And _
      (e.KeyChar <> ".") And _
      (e.KeyChar <> ControlChars.Back)
  End Sub
```

**Listing 1** This is the entire VB.NET class for an extender that provides event handling based on Properties dialog entries (assuming you create it as a component class, which sets the auto-generated code section in the "Component Designer generated code" region). You can create additional extender properties similar to IsNumeric (where the event handler resides in this class), MarkDirty (where the event handler resides in your form), or a combination of the two techniques. The Num\_KeyPress event procedure shows an example of an event procedure in this class and demonstrates how you ignore a user's keypress in VB.NET.

# **The extender component model enables you to add specific extender properties to all, or a subset of, the controls on a form.**

Once you make one or more of these components, you can include them in any form where you want consistent event handling. Simply place the extender component on your form, set the extender properties in the Properties window or through code, and go on to work on more interesting parts of your program.

You'll probably find other uses for the flexibility in VB.NET event handling. Previous event/control array limitations—such as placing a control in only one array, single-dimensional arrays, static groups of controls, a single event procedure for each event, binding

event procedures for the life of the control, and having an event handle only objects of a certain class—are all gone.

You need to let go of old habits and out-of-date practices to take full advantage of VB.NET. Instead of phrasing the problem in an obsolete manner, such as “replacing control array behavior” (which you can also do with this technique), drill down to the underlying problem and explore how VB.NET can solve the problem in potentially better ways. **VSM**

**Kathleen Dollard** is an independent consultant in Fort Collins, Colo. She's a moderator on the DevX forums (<http://news.devx.com>) and a Microsoft MVP. You can reach her at [kathleen@mvp.org](mailto:kathleen@mvp.org).

## **Go Online**

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110DT** Download the code for this article separately. This article's code includes the extender class and a test project that demonstrates its use.

**VS0110DT\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

Microsoft  
**.net**

*Don't settle for less...choose the best!*

**Order  
Online Today!**  
[www.appdev.com](http://www.appdev.com)  
or call 800-578-2062

# **.NET Training**

- CD-ROM Training
- Video Training
- Instructor-Led Classes
- Customized On-Site Classes
- Content Licensing



- ▼ Courses written and taught by industry experts
- ▼ Practical solutions to real world development problems
- ▼ In-depth training in a variety of formats
- ▼ Courses in the latest tools and technologies

## **.NET Courses**

- ▼ Exploring Visual Studio.NET
- ▼ .NET—What It Means to You
- ▼ .NET Framework
- ▼ Visual Basic® 6.0 Migrating to Visual Basic.NET
- ▼ ASP.NET
- ▼ C#

*Many More Coming Soon!*

PRIORITY CODE MG00039

**[www.appdev.com](http://www.appdev.com)**  
**800-578-2062**



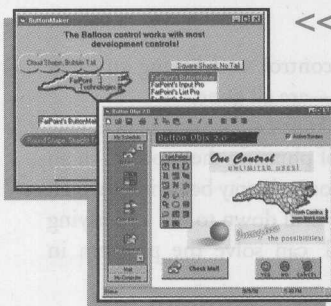
**Instructor-Led Classes • Customized On-Site Classes • CD-ROM Training • Video Training • Content Licensing**

**.NET • Microsoft® Access • ASP • COM+ • HTML • Microsoft® SQL Server™ • Visual Basic® • VBA • Visual FoxPro™ • Visual InterDev™ • XML • XSLT**

**Microsoft® Windows® 2000 • Microsoft® Office 2000**



# AWARD-WINNING COMPONENTS



## << FarPoint's Button Objx 2

Create fully customized buttons, toolbars and custom-shaped containers

FarPoint's Button Objx enhances any web or Windows application. Not only can you replace the Windows button control to create visually enhanced buttons, you can create fully customized active buttons and toolbars as well as custom-shaped containers.

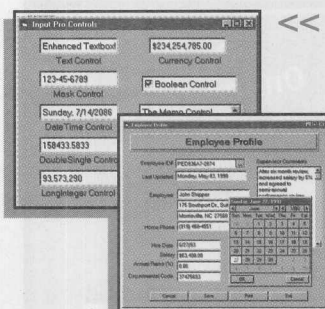
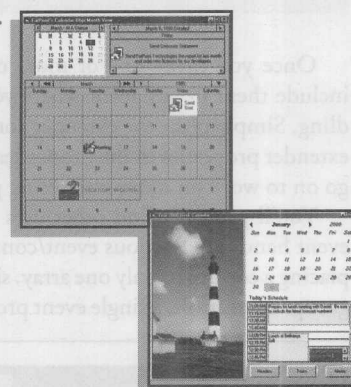
- > Create active-style buttons with or without 3D borders
- > Create a toolbar or tool palette
- > Includes a help balloon (tool tip) for displaying customized text

## FarPoint's Calendar Objx 3

Adding custom calendar and time input to your application has never been this easy

FarPoint's Calendar Objx 3 gives you all the components your development team needs to display or select date and/or time values in any application. By including a calendar control, a clock control, a poster control, and a schedule control, Calendar Objx makes it easy to incorporate robust calendar or schedule features in your program's interface.

- > Choose from multiple calendar display formats
- > Customize all calendar elements
- > Includes five AutoStyles to help you create commonly used calendar looks



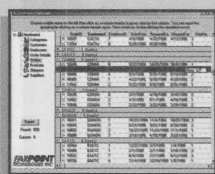
## << FarPoint's Input Pro 3

Full-featured edit components for precise data input

Easily create professional data-entry screens using Input Pro's eight custom controls. Take advantage of the six formatted edit controls to quickly validate data and alert your user as they enter information or as the application programmatically assigns or edits values.

- > Choose from six edit controls (Currency, DateTime, DoubleSingle, LongInteger, Mask, or Text)
- > Create your own appearance or choose from predefined styles
- > Includes a Memo control and Boolean control

## ➔ New Version!



### It's an Enhanced Grid...

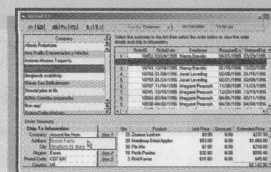
Spread is the best grid to use to present and format tabular or database information in a format your users want.

- > Bind using the most popular data sources or populate in unbound mode
- > Create a grid with up to 2 billion rows by 2 billion columns
- > Provide the most popular data and database formats using 12 cell types
- > Increase data access speed by using the Virtual Data Manager
- > Increased control by validating when and what data gets written to the database

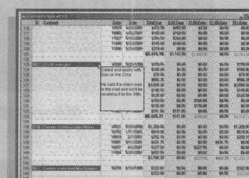
### It's a Spreadsheet...

Spread is the best spreadsheet to use to offer advanced spreadsheet functionality your users need.

- > Import and export spreadsheet files in Microsoft Excel 97/2000 or many



- other formats
- > Export spreadsheet files as XML or HTML files
- > Use one of 105 named functions or create user-defined custom functions
- > Sort with up to 256 sort keys



# FOR THE PROFESSIONAL DEVELOPER

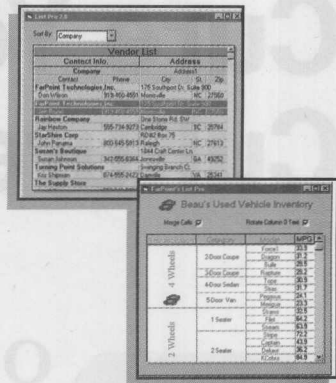
## FarPoint's List Pro 3

*The ultimate data viewer*

FarPoint's List Pro contains the most powerful list box and combo box development components. These extremely robust controls allow you to display up to two billion list items, display single records or items on multiple rows, merge cell text for easier viewing, search for specific list items, or sort list items using unlimited keys.

- > Display data in single or multiple columns
- > Merge column cells that contain identical text
- > Choose an automatic search method

>>



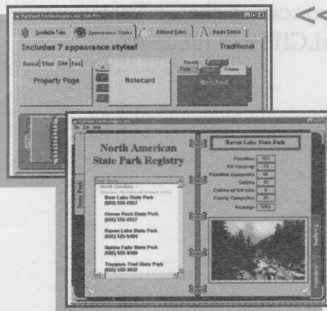
<<

## FarPoint's Tab Pro 3

*Speed, ease of use, and its extensive features make this the "tab of choice"*

Tab Pro is the easiest and quickest way to enhance your application's appearance using a tab metaphor. Whether taking advantage of its eight pre-defined appearance styles or dynamically changing the tab's appearance in response to your user, Tab Pro gives you the most comprehensive tab feature set to customize any application.

- > Choose from eight pre-defined appearance styles
- > Design time support for setting pictures, plus metafile support
- > Create a book chapter effect by assigning multiple pages to one tab



### New Bundles

- >> **Enterprise Developer Toolbox** - MSRP \$849  
Spread, List Pro, Input Pro, Tab Pro, Button Objx, Calendar Objx
- >> **Enterprise Database Pack** - MSRP \$579  
Spread, List Pro, Input Pro

Purchase your copies today by calling  
**800-645-5913**

or download fully functional trial  
versions from our website.

**www.fpoint.com**

**www.fpoint.com**

**FARPOINT**  
TECHNOLOGIES INC

## FarPoint's Spread 3.5

### It's a Report Writer...

*Spread is the best report writer to use to preview and print data in the format your users expect.*

- > Use the Print Preview control to see how your printed spreadsheet will look
- > Print a block of cells, a range of pages, or only the columns and rows that contain data



- > Create custom print configurations by specifying margins, page orientation, borders, gridlines, colors, and shadow effects
- > Print multiple spreadsheets on a single page
- > Add headers and/or footers to your printed pages

### It's So Much More.

Purchase today by calling  
**800-645-5913**

or contact your  
favorite reseller.



MSRP \$399.00

32-bit only

# Cure Your Dates of Culture Shock

by Jonathan Goodyear and Robert Teixeira

## Technology Toolbox

- ☒ VB.NET
- ☒ C#
- ☒ SQL Server 2000
- ☒ ASP .NET
- ☐ XML
- ☒ VB6
- ☒ Other:
  - VB4 and VB5
  - ASP
  - SQL Server 7.0
  - .NET Framework

### Q: Format International Dates

My company plans to deploy an application globally. I experienced many problems with dates in this app during initial testing because of different date formats. How can I format the dates so users see the correct representation for their region?

### A:

The problem here is that different countries use different date formats. Windows recognizes three: m/d/y, d/m/y, and y/m/d. VB makes assumptions about which format to use when it implicitly converts your binary date variables to strings for display, and it doesn't let you specify the one you want. You must use an API call to format the date based on a combination of cultural and regional information. Each language and country setting is represented by a numerical Locale ID (LCID), and you must query the client for what Locale setting to use. If the client

is a Windows application, you can use the `GetThreadLocale` or `GetUserDefaultLCID` API function:

```
Public Declare Function _
    GetThreadLocale Lib "kernel 32" _
    () As Long
```

```
Public Declare Function _
    GetUserDefaultLCID Lib "kernel32" _
    () As Long
```

If your client is a Web browser, you can use this query in ASP:

```
Request.ServerVariables _
    ("HTTP_ACCEPT_LANGUAGES")
```

This returns a Language-Country code string. Refer to the Locale ID chart to translate this code into an LCID (see Resources).

The next step: Use the `VariantChangeTypeEx` API function to change the date into a string (see Listing 1). If you wrote your application using .NET, your job is easier because the .NET Framework comes with built-in cultural/regional support. You can use the .NET Class Library's `CultureInfo` class from either VB.NET or C#. Here's the VB.NET version:

```
Imports System.Globalization
```

```
Public Function DateToString( _
    ByVal dt As Date, ByVal LangCode _
    As String) As String
    Dim ci As New CultureInfo(LangCode)
    Return dt.ToString(ci)
End Function
```

You can download the C# code (Listing A) from the *VSM* Web site; see the Go Online box for details. This

### VB4, VB5, VB6 • Convert a Date to String

```
Public Const S_OK = 0

Public Const VARIANT_NOUSEROVERRIDE = &H4

Public Declare Function VariantChangeTypeEx Lib "oleaut32" _
    (ByRef destVar As Variant, ByRef srcVar As Variant, _
    ByVal LCID As Long, ByVal wFlags As Integer, _
    ByVal VarType As Long) As Long

Public Function DateToString (ByVal dt As Variant, ByVal LCID _
    As Long) As String

    Dim lngRetVal As Long
    Dim strDate As Variant

    lngRetVal = VariantChangeTypeEx(dt, strDate, LCID, _
        VARIANT_NOUSEROVERRIDE, vbString)

    If lngRetVal = S_OK Then
        DateToString = strDate
    Else
        lngRetVal = Win32 Error Code
    End If
End Function
```

**Listing 1** If formatting dates for your global app makes your head spin, use the `VariantChangeTypeEx` API function to change the date into a string. But first you must determine what Locale ID (LCID) you need.

### Resources

Locale ID (LCID) chart: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsmcLCID.asp>



**Figure 1 Render the Results.** You can build this list using a dynamically created Repeater server control. Trap the events for each item in the ArrayList data source to render the same results you'd get if you were to use an actual template in your ASP .NET page.

### My Family:

Jon  
Joy  
CJ  
Ginger  
Cherie

class takes the RFC 1766 standard Language-Country string code automatically, so you don't need to convert it to an LCID. The Date (DateTime in C#) type's ToString() method can use this class to determine the correct string format. —R.T.

## Q: Track Down Bugs

I'm up against a particularly bad and sporadic bug. I can trap errors, but how can I find the series of events that lead up to the error?

**A:**

Having rich error information is nice, but sometimes it's not enough to track down bugs. This is particularly true if you write reusable functionality, especially components. The .NET Framework provides many classes usable by all .NET-compliant languages to aid in program diagnostics. In this case, it sounds like a stack trace is in order. Stack tracing, analogous to opening the Call Stack window in VB6, allows you to view the list of function calls that lead up to the currently running function (in this case, the one with the error).

The Exception class, which has an available instance whenever an error occurs, sports a StackTrace property that's simply a string containing the fully qualified names of all the methods currently in execution. The string also contains the name of the source file and line number where the error occurred if the information is available. If you need a stack trace when there's no error, use Environment.StackTrace instead.

If you need more information, use an instance of a StackTrace class. From the StackTrace object, you can identify the number of frames in the stack and retrieve a StackFrame object for each one.

### VB.NET • Print a Stack Trace to the Console

```
Imports System.Diagnostics

Dim st As New StackTrace()
Dim i As Integer
Dim sf As StackFrame

For i = 0 To st.FrameCount - 1
    sf = st.GetFrame(i)

    ' print line # and function name
    Console.WriteLine( _
        sf.GetFileName().ToString() + " : " & _
        Console.WriteLine(sf.GetMethod().Name)
Next i
```

**Listing 2** Stack tracing, analogous to opening the Call Stack window in VB6, allows you to view the list of function calls that lead up to the currently running function—in this case, the one with the error.

You retrieve the frames in reverse order, starting with the last function pushed. Each frame contains the information for one of the function calls, such as the name of the function, its parameters, and the current line number in the source code of the executing command if the information exists (it's extracted from the debug symbol information). Check out the VB.NET and C# examples of printing a stack trace to the console window (see Listing 2 and download Listing B). Simply copy the code into any function to see it work.

For more power in execution diagnostics, refer to the .NET documentation on the Framework's Trace and TraceListener classes. —R.T.

## Q: Create Server Controls Dynamically

Can you dynamically create server controls, such as the Repeater, that use templates?

**A:**

Why, certainly! You can create most server controls dynamically rather easily. For example, you would represent a typical Label control with this server tag:

```
<asp:Label id="label1" runat="server" />
```

Create label controls dynamically like this:

```
Dim label1 As Label = New Label()
label1.ID = "label1"
Controls.Add(label1)
```

Server controls, such as the Repeater, that leverage templates are a little trickier. Let's give it a shot. I'll present VB.NET code here, but you can find a C# version online (see the Go Online box for details).

First, you need some data for your Repeater to bind to. An ArrayList collection does the trick:

```
Dim al As ArrayList = New ArrayList()
al.Add("Jon")
al.Add("Joy")
al.Add("CJ")
al.Add("Ginger")
al.Add("Cherie")
```

Next, instantiate a Repeater control object and assign its DataSource property to the ArrayList object you just created:

```
Dim r As Repeater = New Repeater()
r.DataSource = al
```

You might not have realized it before, but when you create templates for your server controls, they act as an event handler (of sorts). So you must subscribe to the proper events to handle everything dynamically. You can hook into the Repeater control's ItemCreated event with this code:

```
AddHandler r.ItemCreated, _
    AddressOf repeater_ItemCreated
```

You'll see the implementation of the `repeater_ItemCreated` event a bit later. First, you must fool the Repeater control a bit. The `ItemCreated` event you just subscribed to fires only for the `ItemTemplate` and `AlternatingItemTemplate` templates. These templates are all you need to handle the Repeater's bound data source.

To get the `ItemCreated` event to fire for the `HeaderTemplate`, `FooterTemplate`, and `SeparatorTemplate` templates, specifically assign their respective properties in the Repeater control. These properties require an object that implements the `ITemplate` interface. The `TemplateBuilder` class supports this interface, so you create a new `TemplateBuilder` object instance and assign it to the `HeaderTemplate`, `FooterTemplate`, and `SeparatorTemplate` properties of the Repeater control:

```
Dim tb As TemplateBuilder = New TemplateBuilder()
r.HeaderTemplate = tb
r.FooterTemplate = tb
r.SeparatorTemplate = tb
```

To bind the `ArrayList` to the Repeater control, you need to call its `DataBind` method. Then you render it to the page by adding it to the controls collection of one of the server controls on your ASP .NET page:

```
r.DataBind()
body.Controls.Add(r)
```

In the preceding code, "body" refers to a `Panel` server control on your ASP .NET page:

```
<html>
<head>
<title>Dynamic Repeater - VB</title>
</head>
<body>
<asp:panel id="body" runat="server" />
</body>
</html>
```

You've now seen how to set the gears in motion. Take a look under the hood of the `repeater_ItemCreated` event handler (see Listing 3)—this is where all the rendering decisions are made for each of your virtual Repeater templates. All the Repeater's templates raise the same `ItemCreated` event, so the same event handler will handle them. Determine which template raised the event by interrogating the `ItemType` property of the `RepeaterItemEventArgs` event parameter; you'll find it's a member of the `ListItemType` enumeration. You can now add the proper information to the `RepeaterItem`'s `Controls` collection, which the ASP .NET engine renders to the browser.

Of particular note is the `RepeaterItem` object reference's `DataItem` property, which contains the current item in the Repeater control's data source (in your case, the `ArrayList` object). The `repeater_ItemCreated` event handler gets fired for each item in the data source. It's also fired once for each item so you can handle the `Separator` template, as well as once each for the header and footer templates.

#### VB.NET • Determine Which Template Triggered the Event

```
Private Sub repeater_ItemCreated(Sender As Object, e As _
    RepeaterItemEventArgs)
    'determine which template fired the event
    Select Case e.Item.ItemType
        Case ListItemType.Header
            e.Item.Controls.Add(new LiteralControl("<b>My _
                Family:</b><hr />"))
        Case ListItemType.Item
            e.Item.Controls.Add(new LiteralControl(e.Item. _
                DataItem.ToString()))
        Case ListItemType.AlternatingItem
            e.Item.Controls.Add(new LiteralControl("<i>" & _
                e.Item.DataItem.ToString() & "</i>"))
        Case ListItemType.Footer
            e.Item.Controls.Add(new LiteralControl("<hr />"))
        Case Else 'ListItemType.Separator
            e.Item.Controls.Add(new LiteralControl("<br />"))
    End Select
End Sub
```

**Listing 3** Trap the `ItemCreated` event of the Repeater control to render output to the browser. Determine which template fired the event by interrogating the `RepeaterItemEventArgs`, then construct the proper output.

There you have it: server controls with templates created dynamically (see Figure 1). You can apply the same logic to other server controls that use templates, such as the `DataList` and `DataGrid` server controls. —J.G.

## Q: Can't Get a Date

I'm trying to select items from a table that were created today using the SQL Server `GETDATE()` function, and no results come back. What's wrong?

## A:

The name of SQL Server's `GETDATE()` function is really a misnomer. It returns not only the month, day, and year of the current date, but it also returns the exact system time, down to one three-hundredth of a second. Run this script in Query Analyzer to see what I mean:

```
CREATE TABLE bits
(
    item_name      VARCHAR(10),
    created_dt     SMALLDATETIME
)

INSERT bits VALUES('Item 1',GETDATE())
INSERT bits VALUES('Item 2',GETDATE())
INSERT bits VALUES('Item 3',GETDATE())
```

This simple table contains three rows, each of which contains today's date in its `created_dt` field because you just ran the script. But you won't get any rows back if you run this query:

```
SELECT
    b.item_name
FROM
```

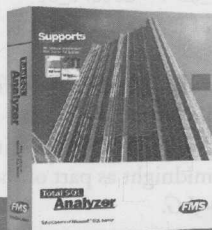
### Drastically Save Development Time

Understand the big picture with top-level summary reports • Understand individual objects with drill-down detail reports • Get up-to-speed quickly on inherited projects

### Dramatically Improve Quality & Performance

Faster, scalable apps with detailed performance analysis • Deliver more reliable apps by detecting errors & omissions • Ensure data consistency and integrity

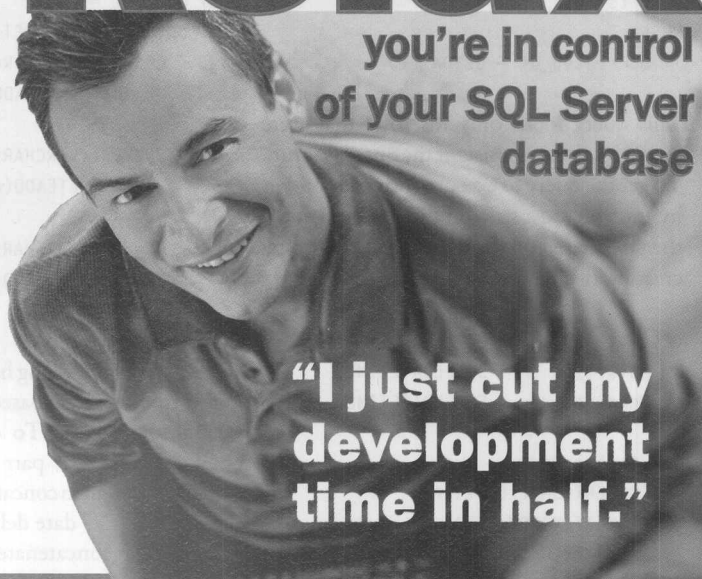
Start relaxing - Call today **Total SQL Analyzer!**



Single Seat \$399  
Five Seat \$1299

# Relax

you're in control  
of your SQL Server  
database



**"I just cut my  
development  
time in half."**

**Start relaxing - contact us today!**

**Call toll free 1-877-367-0449 or visit [fmsinc.com/vsq](http://fmsinc.com/vsq)**

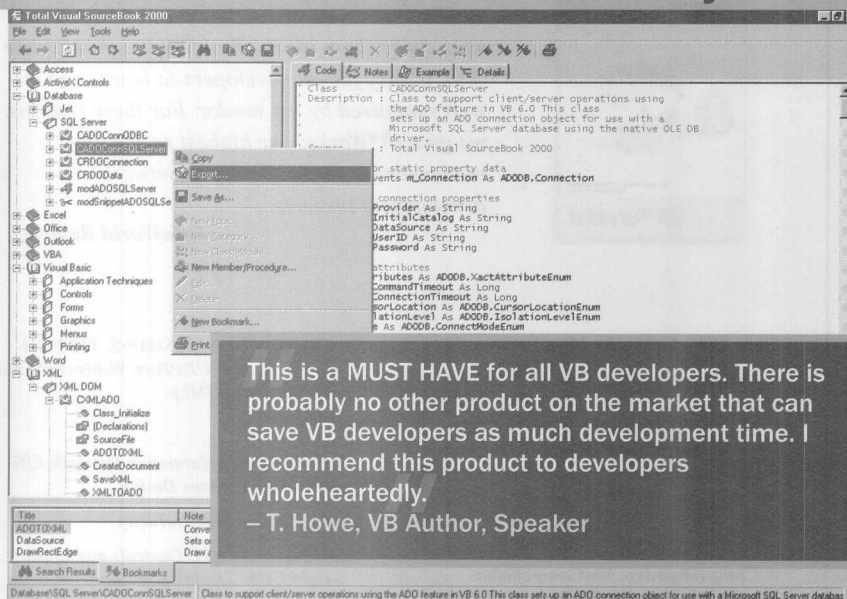
# Improve Productivity!

**Guaranteed! All you need in one VB code library!**

## Total Visual SourceBook 2000

SourceBook saves me hour after hour when I develop. The XML bits have been great for me as I'm fairly new XML, but also the more 'general' code bits give me what I need. And if I don't find exactly what I need, I can always use one or more of the code samples as a template for creating what I want! Thanks!

— C. Thomsen, Microsoft MVP VB



This is a MUST HAVE for all VB developers. There is probably no other product on the market that can save VB developers as much development time. I recommend this product to developers wholeheartedly.

— T. Howe, VB Author, Speaker

Single Seat \$399  
Five Seat \$1299

**Contact us today because you deserve improved productivity!**

**Call toll free 1-877-367-0449 or visit [fmsinc.com/vs](http://fmsinc.com/vs)**



Visit and discover our entire product line for VB  
[sales@fmsinc.com](mailto:sales@fmsinc.com)

# FMS

Since 1986



```
bits b
WHERE
    b.created_dt = GETDATE()
```

This query looks only for rows that were created at the exact instant the query was run. To get the query to execute with your proper intent, you must strip off the time and query the table based on the date alone. Do this by creating two date-range constants:

```
DECLARE
    @start_dt    SMALLDATETIME,
    @end_dt      SMALLDATETIME

SET @start_dt =
    CONVERT(SMALLDATETIME,
    CONVERT(VARCHAR(2),
        MONTH(GETDATE()))
    + '/' +
    CONVERT(VARCHAR(2),
        DAY(GETDATE()))
    + '/' +
    CONVERT(VARCHAR(4),
        YEAR(GETDATE())))
    )
```

```
SET @end_dt =
    CONVERT(SMALLDATETIME,
    CONVERT(VARCHAR(2),
        MONTH(DATEADD(d,1,GETDATE())))
    + '/' +
    CONVERT(VARCHAR(2),
        DAY(DATEADD(d,1,GETDATE())))
    + '/' +
    CONVERT(VARCHAR(4),
        YEAR(DATEADD(d,1,GETDATE()))))
    )
```

What you're doing here is constructing a date from scratch based on its individual component parts. To do that, you must convert each date part from an INT to a VARCHAR, then concatenate them together with a slash "/" date delimiter. Finally, you convert the concatenated string value back into a SMALLDATETIME data type. You didn't specify any time values in the date string, so they'll all be set to zeros.

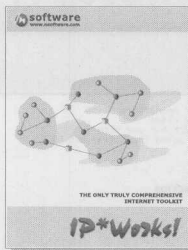
You could use the DATEPART() function as an alternative to the MONTH(), DAY(), and YEAR() functions. The @start\_dt constant contains today's date

with the time stripped off; the @end\_dt constant contains tomorrow's date with the time stripped off. If you use these two constants as date endpoints, you can perform a date range query to get the result you want:

```
SELECT
    b.item_name
FROM
    bits b
WHERE
    b.created_dt >= @start_dt
    AND b.created_dt < @end_dt
```

This query returns all three rows from the "bits" table. Notice you didn't use the BETWEEN predicate. This would've included tomorrow at midnight as part of the allowable values. —J.G.

#### NEW! - VERSION 5.0 - SECURE SOAP, XML, AND A LOT MORE!



*"...IP\*Works! is a network object library that runs everywhere, does all of the things we'd like it to, is easy for all developers to learn, and is well supported by the vendor. For these reasons, we give IP\*Works! our highest rating and recommend it for all Windows network development projects..."*

*InfoWorld Review*

#### Since 1995, In Use By Most Fortune 500s

Initially introduced in 1995, IP\*Works! is actively used by almost all Fortune 500 and Global 2000 companies, as well as thousands of independent software developers worldwide. It has won a number of awards and is highly regarded in the development community for ease of use, robust implementation, and outstanding support.

#### A Comprehensive, No-Compromise, Award-Winning, Professional Solution

With more than 30 components, IP\*Works! includes easy-to-use, standards-based, implementations of all major Internet Protocols: FTP, FileMailer, HTMLMailer, HTTP/S, ICMPPort, IMAP/S, IPDaemon/S, IPInfo, IPPort/S, LDAP/S, MIME, MX, Multicast, NNTP/S, NetClock, NetCode, NetDial, POP/S, Ping, RCP, Rexec, Rshell,

SMTP/S, SNMP, SNPP, SOAP/S, TFTP, Telnet/S, TraceRoute, UDPPort, WebForm/S, WebUpload/S, Whois, XMLp.

#### More Features:

- Razor-Sharp Performance Through Efficient Asynchronous Design
- RSA Certified SSL Security
- Includes Both ATL Controls and Lightweight COM Libraries for ASP
- Unique Security-Control Features Make IP\*Works! The Safest Choice for ASP
- Cross-Platform: Windows, Pocket PC, Linux, Solaris, Java, and beyond!
- .Net Version Coming Soon!

**DOWNLOAD FULLY FUNCTIONAL FREE TRIAL FROM [www.nsoftware.com](http://www.nsoftware.com)**

**Jonathan Goodyear** is the president of ASPSoft ([www.aspsoft.com](http://www.aspsoft.com)), an Internet consulting firm based in Orlando, Fla. He is a Microsoft Certified Solution Developer and holds degrees in accounting and information technology from Stetson University. You can reach him by e-mail at [jon@aspsoft.com](mailto:jon@aspsoft.com) or through his angryCoder eZine at [www.angrycoder.com](http://www.angrycoder.com).

**Robert Teixeira**, a Florida-based consultant, is helping Tech Data's e-commerce department push ahead of the competition, while trading in sleep for writing. E-mail him at [robteixeira@msn.com](mailto:robteixeira@msn.com).

#### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of VSM.

**VS0110QA** Download the code for this article separately. This article's code includes C# code for the first two questions (Listings A and B), VB.NET and C# ASPX files for the dynamically created Repeater control, and the SQL script for stripping the time off of a date in SQL Server.

**VS0110QA\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

DevPartner™  
for Visual Basic

# NOTHING'S EVER EASY

## DETOUR




- Source code analysis
- Automatic error detection
- Performance profiling
- Code coverage
- Automatic runtime error handling

### **Visual Basic is changing. Compuware can help.**

DevPartner for Visual Basic analyzes Visual Basic source code and automatically recommends the changes you should make to ensure future development projects are compatible with Visual Basic.NET. You receive vital information about slow code, performance bottlenecks and code coverage to keep application development on time and on track.

Request your FREE "Moving from Visual Basic 6 to VB.NET" white paper today at [compuware.com/numega](http://compuware.com/numega), or call 1-800-468-6342.

COMPUWARE® 



# Easy E-Mail— It's About Time

Sending e-mail using ASP .NET is so simple, you'll be happy to e-mail—enable your applications.

by A. Russell Jones

## Technology Toolbox

- ☒ VB.NET
- ☐ C#
- ☐ SQL Server 2000
- ☒ ASP .NET
- ☐ XML
- ☐ VB6

**T**he e-mail capabilities in .NET are much more robust than what was available in classic Active Server Pages (ASP). It's difficult to see how sending Simple Mail Transfer Protocol (SMTP) e-mail using ASP .NET could be any easier.

You can send a simple e-mail message with a single line of code. You send an e-mail message by calling the `SmtpMail` object's `Send` command with four `String` parameters: the source account; the destination account; a subject, which appears on the Subject line in the destination account's mail program; and the message body:

```
<!@ Page Language="C#"
AutoEventWireup="false"%>
<%
    System.Web.Mail.SmtpMail.Send (
        "fromaccount@someplace.com",
        "toaccount@somewhereelse.com",
        "This is the subject.",
        "This is the message."
    );
%>
```

The `Send` method is overloaded, so you can create more complex messages. This example uses the `Page_Load` event in a code-behind VB.NET module to send a message with an attachment when the `Page` object loads:

```
Imports System.Web.Mail
' ... standard WebForm
' initialization code here
Private Sub Page_Load(ByVal sender As
```

```
As System.Object, ByVal e As _
System.EventArgs) Handles MyBase.Load
    Dim ms As New MailMessage()
    Dim matt As New MailAttachment _
        ("c:\junk.txt")
    ms.To = "arjl@northstate.net"
    ms.From = "arjiii@hotmail.com"
    ms.Body = "This is a test message."
    ms.BodyFormat = MailFormat.Text
    ms.Subject = "Test message"
```

**Figure 1 Let Users Send an E-Mail.** Users fill out the From, To, Subject, and Message fields on the example WebForm just as they do with any e-mail application. The WebForm also contains an `HtmlInputFile` control (the last control). Note that the File Field adds the Browse button automatically. Clicking on the Send button submits the form.

## Resources

- ASP.NET developer resources: <http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28000440>
- A Guide to .NET: [www.devx.com/dotnet](http://www.devx.com/dotnet)



```
ms.Attachments.Add(matt)
SmtpMail.Send(ms)
End Sub
```

## Let Users Send E-Mail

You can create a WebForm easily that users fill out to send e-mail from your Web site. The userEmail.aspx WebForm example lets users fill out the contents of an e-mail message. (See Listing 1 and Figure 1, and download the code project from the *VSM* Web site; see the Go Online box for details.) Although you would normally set the target address automatically, the example WebForm lets you enter a target address so you can test the program easily.

The WebForm also lets users include a single attachment—but you need to know a couple tricks to make this work. Several standard WebForm Label and TextBox controls are on the form, as well as a standard WebForm button control that submits the data. However, the last control, labeled “Attachment” in Figure 1, isn’t a WebForm server control—it’s an `HtmlInputFile` control. (The `HtmlInputFile` control is actually two controls—a combination of an input text

field and a Browse button.)

It’s convenient to have controls run as server controls, but unfortunately, dropping an `HtmlInputFile` control from the Toolbox onto the WebForm designer disables the `runat` property in the Properties window (see Figure 2). In the example, I set the `HtmlInputFile` control’s name property to `txtAttachment`. Even worse, Visual Studio doesn’t add the control to the list of controls it creates for the page in the code-behind module. Don’t worry. You can fix both problems manually.

With the WebForm in design mode, click on the HTML button to view the WebForm HTML source code. Find the `HtmlInputFile` control (look for an `<INPUT>` tag that has the `type=“file”` attribute) and add the attribute `runat=“server”`:

```
<INPUT id="txtAttachment" runat="server"
style="Z-INDEX: 114; LEFT: 108px;
WIDTH: 522px; POSITION: absolute;
TOP: 330px; HEIGHT: 22px" type=
"file" size="67">
```

### HTML, VB.NET (beta 2) • Users Can Fill Out Message Contents

```
<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="userEmail.aspx.vb"
Inherits="dotNetEmail.userEmail"%>
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<title></title>
<meta content="Microsoft Visual Studio.NET 7.0"
name="GENERATOR">
<meta content="Visual Basic 7.0"
name="CODE_LANGUAGE">
<meta content="JavaScript"
name="vs_defaultClientScript">
<meta content="http://schemas.microsoft.com/
intellisense/ie5" name="vs_targetSchema">
<LINK href="Styles.css" type="text/css"
rel="stylesheet">
</HEAD>
<body bgColor="#ffffff"
MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server"
enctype="multipart/form-data">
<asp:label id="lblDirections" style="Z-INDEX:
101; LEFT: 36px; POSITION: absolute; TOP:
25px; Design_Time_Lock: True" runat="server"
Width="632px" Height="19px" Design_Time_Lock=
"True" CssClass="directions">Label</asp:label>
<asp:textbox id="txtMessage" style="Z-INDEX: 109;
LEFT: 109px; POSITION: absolute; TOP: 152px"
runat="server" Width="438px" Height=
"161px"></asp:textbox>
<asp:textbox id="txtSubject" style="Z-INDEX: 108;
LEFT: 109px; POSITION: absolute; TOP: 119px"
runat="server" Width="437px" Height=
"24px"></asp:textbox>
<asp:textbox id="txtTo" style="Z-INDEX: 107;
LEFT: 109px; POSITION: absolute; TOP: 86px"
runat="server" Width="437px" Height=
"24px"></asp:textbox>
<asp:label id="lblMessage" style="Z-INDEX: 105;
LEFT: 36px; POSITION: absolute; TOP: 150px;
Design_Time_Lock: True" runat="server"
Design_Time_Lock="True" CssClass="smalllabel">
Message</asp:label>
<asp:label id="lblSubject" style="Z-INDEX: 104;
LEFT: 36px; POSITION: absolute; TOP: 121px;
Design_Time_Lock: True" runat="server"
Design_Time_Lock="True" CssClass="smalllabel">
Subject</asp:label>
<asp:label id="lblTo" style="Z-INDEX: 103; LEFT:
36px; POSITION: absolute; TOP: 86px;
Design_Time_Lock: True" runat="server"
Width="60px" Height="19px" Design_Time_Lock=
"True" CssClass="smalllabel">To</asp:label>
<asp:label id="lblFrom" style="Z-INDEX: 102;
LEFT: 36px; POSITION: absolute; TOP: 55px;
Design_Time_Lock: True" runat="server" Width=
"60px" Height="19px" Design_Time_Lock="True"
CssClass="smalllabel">From</asp:label>
<asp:textbox id="txtFrom" style="Z-INDEX: 106;
LEFT: 109px; POSITION: absolute; TOP: 55px"
runat="server" Width="437px" Height="24px">
</asp:textbox>
<asp:button id="btnSend" style="Z-INDEX: 110;
LEFT: 438px; POSITION: absolute; TOP: 359px"
accessKey="S" runat="server" Width="108px"
Height="24px" Text="Send" BorderWidth="3px"
BorderStyle="Outset"></asp:button>
<asp:label id="lblAttachment" style="Z-INDEX:
111; LEFT: 28px; POSITION: absolute; TOP:
328px" runat="server" Width="61px"
Height="19px">Attachment</asp:label>
<INPUT id="txtAttachment" runat="server"
style="Z-INDEX: 114; LEFT: 108px; WIDTH:
522px; POSITION: absolute; TOP: 330px; HEIGHT:
22px" type="file" size="67">
</form>
</body>
</HTML>
```

**Listing 1** The generated HTML for the userEmail.aspx WebForm contains several WebForm controls and one `HtmlInputFile` control (the last control). If you want to access an `HtmlFileInput` control with code after dragging it onto a WebForm, you must add the `runat=“server”` attribute manually to the generated HTML.

Next, navigate to the code-behind module (called `userEmail.aspx.vb` in the example) and add this line:

```
Protected txtAttachment As   
System.Web.UI.HtmlControls.HtmlInputFile
```

Declaring the control lets you access the control's properties in the code-behind module just as you can any other WebForm control. Note that you only need to do this if you're using code-behind modules—it's not necessary to declare the control if you're using embedded code.

You need to collect the field values from the form to send the message. When a user clicks on the Send button, the form posts the entered data back to the server and fires the `btnSend_Click` event (see Listing 2).

Despite the simplicity of sending e-mail, messages can fail. For example, the SMTP server refuses to send an e-mail if the sender's

address doesn't belong to a valid domain. Although the examples that accompany this article don't include error traps, you can and should trap these errors in production code.

You aren't limited to sending plain text messages; you can send e-mail formatted in HTML too. For example, you can create an alternate version of the WebForm that scans the message body for HTML tags (see Listing 3). When it finds them, it sends the e-mail formatted as HTML; otherwise, it sends the message as plain text.

## Scan Messages for Content

Other than the added check to see whether the message contains HTML, the code is identical to the previous version. However, the check serves as an introduction to the power of regular expressions in .NET and shows how you might scan messages for specific content—perhaps for routing to an appropriate contact.

The gateway to regular expression matching is the `Regex` object, which scans a `String` for pattern matches. You specify the pattern to match with an arcane syntax consisting of a string of symbols and characters. For example, in Listing 3, the pattern ("`<\w.*>`") matches any HTML tag, which means the `Regex` object matches a character

### VB.NET (beta 2) • Send a Message With an Attachment

```
Imports System.Web.Mail
' ... standard WebForm
' initialization code here
Private Sub btnSend_Click(ByVal sender As   
System.Object, ByVal e As   
System.EventArgs) Handles btnSend.Click
    Dim mm As New MailMessage()
    Dim mattach As MailAttachment
    Dim fileTitle As String
    Dim aPos As Integer

    ' set the MailMessage properties
    mm.To = txtTo.Text
    mm.From = txtFrom.Text
    mm.Subject = txtSubject.Text
    mm.Body = txtMessage.Text

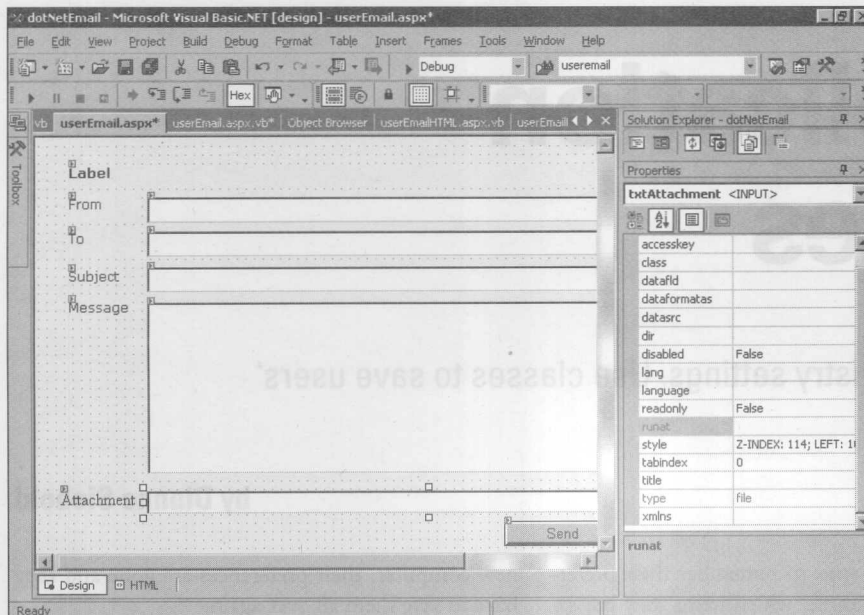
    ' get the attachment, if present
    If Not IsNothing   
(txtAttachment.PostedFile) Then
        ' get the file title without the path
        fileTitle = txtAttachment.   
PostedFile.FileName
        fileTitle = fileTitle.Substring   
(fileTitle.LastIndexOf("\") + 1)
        ' save the posted file
        txtAttachment.PostedFile.SaveAs   
( "c:\temp\" & fileTitle)
        ' create a MailAttachment object
        mattach = New MailAttachment   
( "c:\temp\" & fileTitle)
        ' add the attachment to the MailMessage
        mm.Attachments.Add(mattach)
        ' delete the temporary file
        System.IO.File.Delete   
( "c:\temp\" & fileTitle)
    End If
    SmtpMail.Send(mm)
End Sub
```

**Listing 2** This alteration of the `btnSend_Click` event code sends a message with an attachment, if one is present. It retrieves the uploaded file data from the `HtmlFileInput` control's `PostedFile` property and saves it to a temporary file on the server. Then it creates a new `MailMessage` object, passing the name of the temporary file as a parameter to the `MailMessage` constructor.

### VB.NET (beta 2) • Scan for HTML Tags

```
Private Sub btnSend_Click(ByVal sender As   
System.Object, ByVal e As   
System.EventArgs) Handles btnSend.Click
    Dim mm As New MailMessage()
    Dim mattach As MailAttachment
    Dim fileTitle As String
    Dim aPos As Integer
    Dim rx As Regex
    Dim rxmatch As Match
    mm.To = txtTo.Text
    mm.From = txtFrom.Text
    mm.Subject = txtSubject.Text
    mm.Body = txtMessage.Text
    If Not IsNothing(txtAttachment.   
PostedFile) Then
        fileTitle = txtAttachment.PostedFile.   
FileName
        If fileTitle.Length > 0 Then
            fileTitle = fileTitle.Substring   
(fileTitle.LastIndexOf("\") + 1)
            txtAttachment.PostedFile.SaveAs   
( "c:\temp\" & fileTitle)
            mattach = New MailAttachment   
( "c:\temp\" & fileTitle)
            mm.Attachments.Add(mattach)
            System.IO.File.Delete("c:\temp\" &   
fileTitle)
        End If
    End If
    rx = New Regex("<\w.*>")
    rxmatch = rx.Match(mm.Body)
    If rxmatch.Success Then
        mm.BodyFormat = MailFormat.Html
    Else
        mm.BodyFormat = MailFormat.Text
    End If
    SmtpMail.Send(mm)
End Sub
```

**Listing 3** This altered version of the `btnSend_Click` event code checks to see if the message body submitted by the user contains at least one HTML tag. If so, it sends an HTML-formatted message; otherwise, it sends a plain text message.



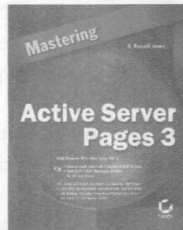
**Figure 2 Change the Value Manually.** When you place an `HtmlInputFile` control on the WebForm design surface, Visual Studio disables the `runat` property. However, you can change the value manually by clicking on the HTML tab at the bottom of the design window to access the HTML content (see Listing 3).

pattern that begins with a left angle bracket followed by any word character (`\w`, which matches `a-z`, `A-Z`, or `0-9`), followed by 0 or more additional characters of any type (`.*`), and finishes with a right angle bracket. The pattern shown in this simple example won't match a VB not-equal symbol (`<>`) or angle brackets that enclose only white space, but it is by no means robust enough for production-quality code; for example, the pattern shown matches XML tags and invalid HTML tags as well as valid HTML tags. Rather than trying to determine the content type automatically, most e-mail programs let users select whether they want to send messages as HTML or text.

The `Regex` object returns a `Match` object. The `Match.Success` property has a value of `True` if any matches occurred. The `Match` object contains far more information than a simple Boolean value, but this example doesn't need any advanced `Match` object functionality—the WebForm sends the message as HTML if *any* match occurs. Use the `MailMessage` object's `BodyFormat` property to control the format. The `BodyFormat` property accepts one of two enumeration values: `MailFormat.Html` or `MailFormat.Text`.

In the future, the .NET Framework will undoubtedly be able to send e-mail messages in other ways, but right now, it's enough that sending mail is finally both simple and built into the framework. **VSM**

**A. Russell Jones**, Ph.D., is a senior Web developer for DevX. He's a former reptile keeper and professional musician who now composes computer applications. He's the author of *Visual Basic Developer's Guide to ASP and IIS* and *Mastering Active Server Pages 3* (both published by Sybex). Reach him at [arj1@northstate.net](mailto:arj1@northstate.net).



### Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110AN** Download the code for this article separately. This article's code includes code to create a WebForm that users fill out to send e-mail from your Web site.

**VS0110AN\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).

In one week,  
not one year!

It can't be this easy... but it is.

True OLE DB Provider  
SDK

**ComponentOne**

[www.componentone.com](http://www.componentone.com)

1.800.858.2739 • 1.888.228.4839

© 2001 ComponentOne LLC. All rights reserved.  
All product names are owned by their respective holders.



# Save Application Preferences

Say goodbye to INI files and Registry settings. Use classes to save users' preferences to your database.

by Dianne Siebold

## Technology Toolbox

- ☐ VB.NET
- ☐ C#
- ☒ SQL Server 2000
- ☐ ASP .NET
- ☐ XML
- ☒ VB6
- ☒ Other:
  - ActiveX Data Objects (ADO) 2.5

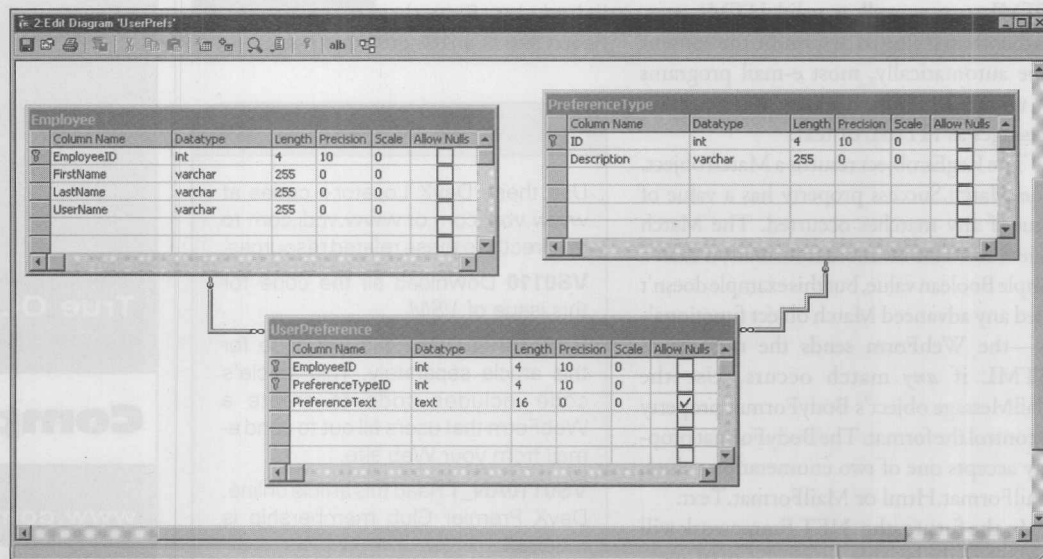
Users like your apps to remember their preferences—for example, which items they last selected in a listbox control or the widths of the columns in a grid. This means you not only have to write the code to save and retrieve these preferences, but you need to figure out where to store this information. In this column, I'll show you how to use SQL Server to store user preference data (you can also modify this approach easily to work with any SQL database). This method is scalable: It works well for small, client/server applications as well as large ones.

Earlier versions of Windows used INI files to save preferences. Now the Registry is generally the preferred location for storing user settings, but it has some drawbacks. You can use the Visual Basic SaveSetting command and the GetSetting function for saving and retrieving Registry entries. However, the Registry saves preferences locally, so if users get a

new computer, their preferences are gone and they have to save them all over again.

There's a better way to save your application's user preferences to a database: Use two simple classes that handle saving and retrieving the preferences. There's no limitation to the number and types of settings your application can save because the data is stored in a table. I'll show you the architecture used to save and retrieve user preferences and how to implement it. The sample application demonstrates how to use these classes by saving and loading both the items selected in a listbox, and the width of columns in a grid (download the code from the *VSM* Web site; see the Go Online box for details).

There are several benefits to saving user settings to a database. All user settings are saved centrally, so you need to make administrative changes only in one location instead of at each user's computer. Prefer-



**Figure 1 Store Preferences in the UserPreference Table.** You need three tables to implement saving preferences to a database. The UserPreference table contains each user's preferences. This table has a foreign key to the PreferenceType table and the Employee table.

ences are global: They're available to users no matter which computer they use to run your application. This approach is also scalable. It works in a simple client/server environment, as well as a distributed environment. User preferences are backed up regularly with the database, there's no limitation to the number and type of user preferences you can save, and you can use this approach with any database.

## Create the Classes

You'll take a bottom-up approach to creating this application, so start by looking at how to store the user preferences in the database. Add three tables to your database by running the SQL code in the readme.txt file in the sample app: Employee, UserPreference, and PreferenceType. The primary table is UserPreference, which stores each user's preference values (see Figure 1). This table has a foreign key to the Employee table on the EmployeeID column and a foreign key to the PreferenceType table on the PreferenceTypeID column. A column of Text data type stores the data that makes up the preferences. The SQL code also adds some data to the Employee and PreferenceType tables and adds a stored procedure to the current database.

Take a look at the classes in the test project. The UPPreferences.Prefs class can be client-side or Microsoft Transaction Server (MTS)/COM+, and it handles the database interaction of saving and retrieving the preferences. To enable this class to run in MTS or COM+, you need to set the MTSTransactionMode property to a value greater than No Transactions and put the appropriate SetCommit and SetAbort statements in the methods. The client passes the preferences into the Prefs class in a recordset designed to be marshaled efficiently.

The second class, called UPPropertyBag, is a generic client-side object that handles parsing the various elements of the preferences. This property bag object is similar to the VB property bag, but with some improvements. For example, the property bag passes its contents as a string instead of a byte array. Each element or preference in the string has a key and a value, so the grid width preferences for each column look like this:

```
"Column0""900""Column1""1200""Column2"
"700"
```

Just like the VB property bag, the WriteProperty method saves values to the property bag, and the ReadProperty method retrieves them. The Contents property gets and sets the property bag contents, which the client passes as a string (unlike the VB property bag, which requires them as a byte array).

Take a look at how these objects work together to retrieve and

## VB6 • GetUserPreferences Returns a Single User's Preferences

```
Private Function GetUserPreferences() As Boolean

Dim PrefsRS As ADODB.Recordset
Dim AppPropertyBag As UPPropertyBag.PropertyBag

'The preferences for this user are returned in
'a recordset.
If Not mPreferences.GetUserPreferences _
(mEmployeeID, PrefsRS) Then Exit Function

'Loop through the recordset and save each
'preference in its own property bag.
PrefsRS.MoveFirst
Do Until PrefsRS.EOF
Set AppPropertyBag = New _
UPPropertyBag.PropertyBag
If Not IsNull _
(PrefsRS("PreferenceText").Value) Then
AppPropertyBag.Contents = _
PrefsRS("PreferenceText")
End If

'Add each property bag to a form-level
'collection.
mUserPreferences.Add AppPropertyBag, _
CStr(PrefsRS("DescPreferenceType"))

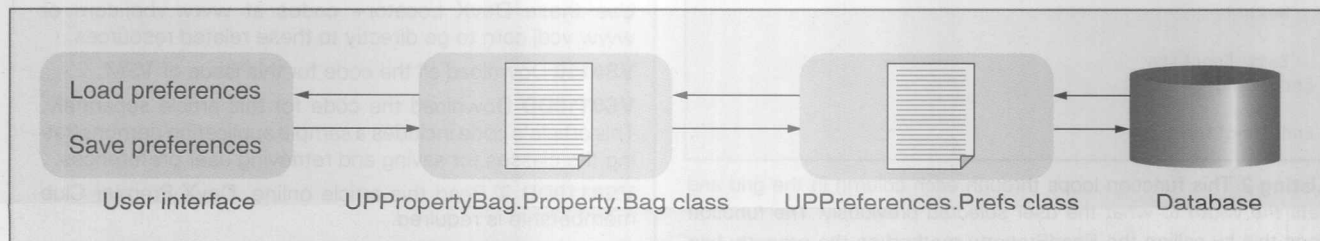
Set AppPropertyBag = Nothing
PrefsRS.MoveNext
Loop

GetUserPreferences = True
End Function
```

**Listing 1** The GetUserPreferences function calls a method of the same name to return the preferences for a single user in a recordset. The function then adds the preferences to a property bag, and stores each property bag object in a collection.

save settings in a user interface. The test application has a grid and a listbox. You want to save the items the user last selected in the listbox and the widths of the grid columns, so when users open this form, their previous settings are loaded. The Form\_Load event makes a call to a function called GetUserPreferences and retrieves all the preferences for the specified user with this query:

```
SELECT up.EmployeeID, pt.ID
PreferenceTypeID,
pt.Description DescPreferenceType,
up.PreferenceText
FROM PreferenceType pt
```



**Figure 2 Save User Preferences.** Instead of saving program settings to individual users' machines, this approach saves them to the database using the UPPropertyBag.PropertyBag and UPPreferences.Prefs classes.

```
LEFT OUTER JOIN UserPreference up
ON up.PreferenceTypeID = pt.ID AND
up.EmployeeID = 108
LEFT OUTER JOIN Employee e ON
e.EmployeeID = up.EmployeeID
AND e.EmployeeID = 108
```

Notice that the query selects from the PreferenceType table and has an outer join to the UserPreference table. This creates a recordset that contains a row for each preference type and the user's preferences for that type, if there are any. The client saves the preference types that a user has previously saved preferences for into property bag objects and adds them to a form-level collection called mUserPreferences (see Listing 1). You access a particular property bag by creating a form-level variable for a specific bag, then setting it to the appropriate bag in the mUserPreferences collection using the preference type description:

```
Private mGridPrefsBag As _
UPPropertyBag.PropertyBag

Set mGridPrefsBag = mUserPreferences _
("TestForm.GridPreferences")
```

Now that you have a reference to the grid preferences property bag, use it to set the grid column widths to the same values as the last time the user opened this form. Do this by using the ReadProperty method to query the values from the mGridPrefs property bag (see Listing 2).

## Save Screen Settings

The code that saves the preferences works in reverse order and saves the user preferences to the property bag first (see Figure 2). Using the grid preferences as an example, the code in the Form\_Unload event loops through each column in the grid and writes the column widths to the property bag. The key for each

value is the string "Column" plus the index of the column (Column0, Column1, and so on):

```
For i = 0 To (MSFlexGrid1.Cols - 1)
mGridPrefsBag.WriteProperty _
"Column" & CStr(i), CStr _
(MSFlexGrid1.ColWidth(i))1
Next i
```

## SaveUserPreferences

Once you save the values in the property bag, the SaveUserPreferences function saves them to a recordset (download Listing 3). The SaveUserPreferences function then passes this recordset into the Prefs class's SaveUserPreferences method, where it saves the preferences for this user to the UserPreference table. The stored procedure checks to see if the specified user has a record for the preference type already. If so, it's updated; if not, a new record is inserted.

This is where this method's flexibility comes in. You can save preferences in two different ways. First, you can create an entry in the PreferenceType table for each control in a screen, which is how it's done in the sample application. You have separate entries such as TestForm.GridPreferences, TestForm.ListBoxPreferences, and so on. Second, you can save all the preferences for the form's controls in a single string. In that case, you have one entry in the PreferenceType table (TestForm.Preferences, for example) and one property bag in the application.

If you need to implement the ability to save user preferences in your application, or if you want to move from INI files or Registry settings, this might be the solution for you. Not only will this method make your job as a developer easier, but the ability to save preferences will make your application much more polished and professional. **VSM**

*The code for this article is based on application code written by Cimarron Software (www.cimarronsoftware.com). Thanks to Richard Lidstrom and Goran Stijacic for letting me use it.*

### VB6 • Set the Width

```
'Sets the column widths of the grid columns
'based on what was previously selected.
Private Function LoadGridPrefs() As Boolean

On Error GoTo ErrorHandler

Dim i As Integer

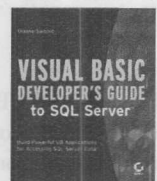
For i = 0 To (MSFlexGrid1.Cols - 1)
MSFlexGrid1.ColWidth(i) = _
mGridPrefsBag.ReadProperty("Column" & CStr(i))
Next i

Exit Function
ErrorHandler:

End Function
```

**Listing 2** This function loops through each column in the grid and sets the width to what the user selected previously. The function does this by calling the ReadProperty method on the property bag and passing the key of the desired value—in this case, the word "Column" with the column index appended to it.

**Dianne Siebold** is a consultant specializing in Visual Basic and SQL Server programming. She is a regular contributor to *Visual Studio Magazine* and the author of the *Visual Basic Developer's Guide to SQL Server* (Sybex). Reach her by e-mail at dsiebold@earthlink.net.



## Go Online

Use these DevX Locator+ codes at [www.vbpj.com](http://www.vbpj.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

**VS0110** Download all the code for this issue of *VSM*.

**VS0110DD** Download the code for this article separately. This article's code includes a sample application demonstrating the classes for saving and retrieving user preferences.

**VS0110DD\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).



# Develop Better and Faster TODAY!

# VSLive!™

Visual Studio Developer Conference  
**ORLANDO ♦ OCTOBER 9-14**

**Register Now!**

**800-848-5523**

**[www.vslive.com/vsm](http://www.vslive.com/vsm)**

**VBITS**  
Visual Basic INSIDE™ TECHNICAL SUMMIT

**SQL**  
ZTheMax

**Visual C++**  
Developers Conference

**Over 180 Hours of ALL NEW  
Technology for the .NET Platform,  
C#, XML, T-SQL and more!**

- ◆ FULL BONUS DAY ON .NET!
- ◆ Explore the Bleeding Edge of Web Development with VB, ASP.NET, C# and XML
- ◆ Plunge into Windows XP and Web Services
- ◆ Exploit Data Access with ADO.NET, SOAP, and T-SQL
- ◆ PLUS: HailStorm Architecture Unleashed

**Call 800-848-5523 to Register**  
**Visit [www.vslive.com/vsm](http://www.vslive.com/vsm)**  
**for More Details**

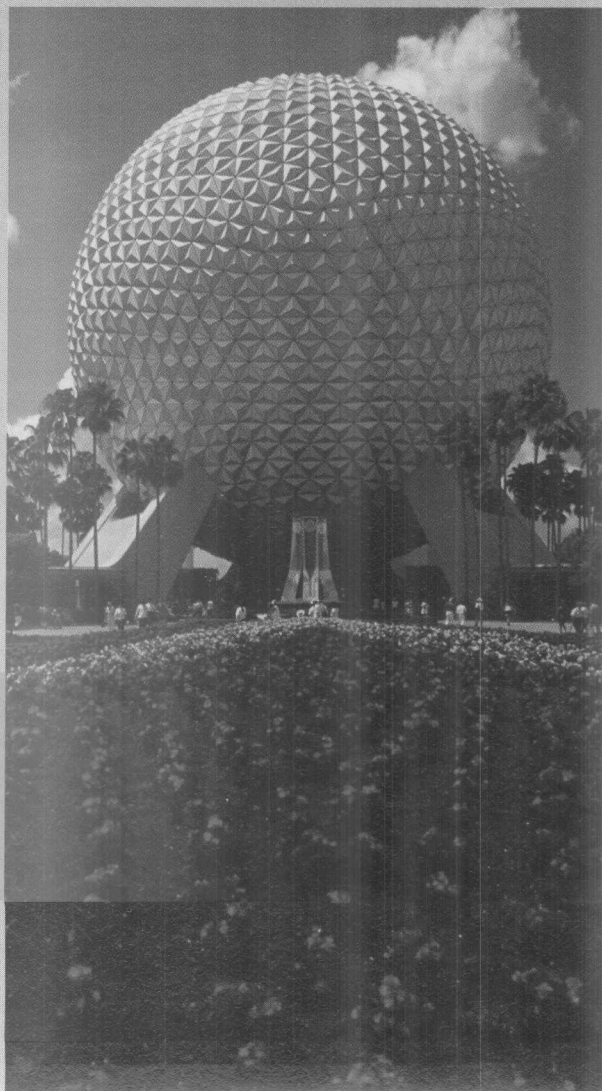
**Microsoft**

**msdn**

**FTP** FAWCETTE  
TECHNICAL  
PUBLICATIONS

**Visual Studio**  
Magazine

Microsoft, MS, MSDN, SQL Server, Visual Basic, Visual C++, Visual Studio, and Windows are registered trademarks of Microsoft Corporation. VBITS and Visual Studio Magazine are registered trademarks and Visual C++ Developers Conference, XML Magazine and SQL2TheMax are trademarks of Fawcette Technical Publications, Inc. Visual Studio is used by Fawcette Technical Publications, Inc. under license from Microsoft. All other trademarks are property of their respective owners.



# Develop an ATL Web Service

VS.NET's new Web Services programming model provides a robust method to deliver component services across the Web.

by Bill Wagner

## Technology Toolbox

- ☐ VB.NET
- ☒ C#
- ☐ SQL Server 2000
- ☐ ASP .NET
- ☐ XML
- ☐ VB6
- ☒ Other:

Visual C++ .NET  
Windows 2000 Server

The Web Services architecture is one the most innovative new development technologies in the Microsoft .NET Framework. It enables you to build components that can deliver their services across the Internet to client programs (see the sidebar, "Serving Up Web Services").

Visual Studio .NET (VS.NET) provides many tools to create Web Services—*Visual C++ Developers Journal* readers learned how to build one using C# (see Resources). Here, you'll learn to use ATL Server's Web Services tools to build a Web Service using C++. You'll also build a C# client to take advantage of the Web Service.

As an example, I'll use data from the U.S. Census Bureau (see Resources). This site could really use

some Web Services! Browse the site and you'll find a wealth of collected data in HTML tables or PDF files, but you can use the data only as formatted by the Census Bureau's Web server, and you can't make comparisons or calculations using the data directly. Your simple Web Service will deliver sample data from the Census Bureau to client applications, and your C# client will display this data in a format you determine.

You'll provide three Web methods in your Web Service. The simplest, `GetStatePopulation()`, lets a client retrieve the most recent population tally for a given state. The second, `GetAllPopulation()`, lets a client retrieve a data structure that contains the most recent population numbers for all the states. The

## Serving Up Web Services

Web Services are a simple concept. A Web Service is simply a component that accepts requests delivered as XML files using either HTTP or HTTP POST commands, and responds to the requests by delivering an XML file back to the client. VS.NET provides several tools that let you concentrate on your particular problem rather than on the transport protocol or the XML coding.

The previous Web application model—a thin client model—provided as much computational power on the server as possible, with the client responsible only for data display. This worked fine when the data display was textual, or static. But new Web-based applications demand more processing to format data and present it. Most users have ample processing power on their desktops, but many modern applications work with huge amounts of data. It's completely unreasonable to expect users to have the amount of storage or processor capability associated with large enterprise servers.

Enter Web Services. Web Services provide a mechanism for the server computers and client

computers to perform the task each is best suited for. Servers can store large amounts of data and do computations on them. Clients can execute the best presentation for that data and provide a rich environment for users.

You can also use the Web Service model to allow two different Web sites to cooperate to provide a better user experience. As an example, consider online retailers and shipping companies. After you place an order, most online retailers provide some mechanism for you to query the order status. If your order has shipped already, they instead provide you with the tracking number and direct you to the shipping company's Web site. Once there, you enter the tracking number and get the new status information. If the shipping company provided a Web Service for other Web sites to track shipments, the online retailer could send a request to the shipping company's Web site to retrieve the shipping information as part of the response to your query about your order, then present you with all the information in one page.

## Resources

- Middle Tier, "Create a Web Service With VS.NET," by Alan Gordon [*Visual C++ Developers Journal* February 2001]
- XML Developer Center: <http://msdn.microsoft.com/xml/default.asp>
- SOAP Developer Resources: <http://msdn.microsoft.com/soap/default.asp>
- Web Services Developer Resources: <http://msdn.microsoft.com/webservices/default.asp>
- U.S. Census Bureau: [www.census.gov](http://www.census.gov)



third, `GetTimeSeries()`, lets a client retrieve all the historical population data from a single state.

Start VS.NET and select New Project from the File menu, then select the Win32 Projects folder from inside the Visual C++ Projects Folder. Select ATL Server Web Service Project, and name the project "CensusServer" (see Figure 1). Most of the defaults for this project are correct. I find it easier to build a combined DLL for an ATL Web Service, so go to the Project Settings tab and check the Generate Combined DLL box for simpler deployment. Your project creates one DLL that holds the ISAPI extension and the Web Service; it creates two separate DLLs when this box is unchecked.

Now define the interface, `ICensusServerService`, for the three methods you'll implement. The ATL Server AppWizard provides boilerplate code that includes the canonical "Hello World" Web Service method; you'll remove this code and go through a two-step process to add the interface definition for the `CensusServer` methods (see Listing 1). First, provide the structures for the nodes retrieved when a client asks for multiple data points. These structures define the XML nodes the Web Service returns as part of the methods. The first node defines a state identifier and a population; the second, a year and a population.

Second, define the `GetStatePopulation()`, `GetAllPopulation()`, and `GetTimeSeries()` Web Service methods. Add the first method's definition to your sample now (see Listing 2)—you'll add the next two methods later.

If this were a production Web Service connected to real Census data, the method implementation would use a large database to find the answers. This is a sample application, so you'll add code to store the information this service provides. Use a C++ standard library map to store the state/population pairs by adding the necessary includes at the top of the `CensusServer.h` file:

```
#include <map>
#include <string>
using namespace std;
```

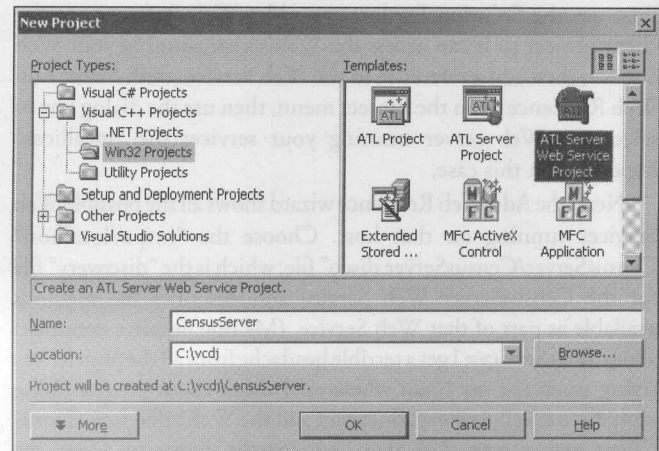
Next, add member variable declarations to store the map between state abbreviations and populations:

```
private:
    static map <string, long >
        CurrentPopulationStats;
```

Finally, add a constructor to initialize this data structure with the information containing the pertinent data. Download the code from the *VSM* Web site to see the boilerplate code that fills the map (see the Go Online box for details). Finally, add the method implementation for the `GetStatePopulation()` Web Service method (see Listing 2). Now build the service and run it to display its Web Service *contract*, a description that lets client programs build proxies to communicate with it (see Figure 2).

## Build the Client

Next, build a client to use this new Web Service—call it `Census Client`. Select New Project from the File menu, then choose C# Windows Application. Add the client application to the current



**Figure 1 Create an ATL Server Web Service.** VS.NET makes it easy to get started with an ATL Server Web Service. It creates the Web Application and ISAPI Extension DLLs, and even adds deployment support.

## VC++.NET • Define the Web Service Interface

```
// The XML Node for a state/population pair
[ export ]
struct StatePopulation {
    BSTR state;
    int population;
};

// The XML Node for a point in the time series
[ export ]
struct YearPopulation {
    int year;
    int population;
};

// ICensusServerService - Web service
// interface declaration
//
[
    uuid("5C346C5D-54FD-471E-81FE-5D03551A5010"),
    object
]

__interface ICensusServerService
{
    [id(1)] HRESULT GetStatePopulation
        ([in] BSTR stateID, [out, retval] long*
            population);
    [id(2)] HRESULT GetAllPopulation
        ([in,out] int *count,
        [out,size_is(*count)] StatePopulation**
            pStates);
    [id(3)] HRESULT GetTimeSeries ([in] BSTR
        stateID,
        [in,out] int *count,
        [out,size_is(*count)] YearPopulation**
            pYears);
};
```

**Listing 1** This code defines the `CensusServer` Web Service interface, which contains two node definitions and three methods. The compiler uses the attributes to produce the type library for the Web Service.



solution to simplify working with both parts of the application. First use the Solution Explorer to add a Web Reference to the client project so it can access the Web server running your Web Service, then add a reference to the Web Service itself. Select Add Web Reference from the Project menu, then use the dialog box to select the Web server running your service: the "localhost" hyperlink, in this case.

Next, the Add Web Reference wizard shows all the possible Web Services running on that host. Choose the "http://localhost/CensusServer/CensusServer.disco" file, which is the "discovery" file for the CensusServer Web Service and describes the methods available as part of that Web Service. (My one negative comment about Web Services: I get a terrible headache from all the punch lines trying to escape my brain whenever I think about programming with disco files.) Anyway, once you add the Web reference, Census Client gets a new class that contains the proxy to access the CensusServer Web Service.

Add controls to the form to test your first interface. Add a combo box to pick the state, a button to get the current population, and a textbox to display the population (see Figure 3). Then add the code to access the Web Service. First declare a member variable in the Form object for the Web Service proxy, then create the object in the Form's constructor:

```
private localhost.CensusServerService
    cs;
cs = new
    localhost.CensusServerService();
```

Finally, add a click handler to display the population of the selected state:

```
protected void button1_Click (object
    sender, System.EventArgs e)
{
    String s = comboBox1.Text;
    int pop =
        cs.GetStatePopulation (s);
    textBox1.Text=pop.ToString();
}
```

That's all there is to it. Try it yourself: Build the project, run it, select a state from the combo box, and hit the button. The Web Service returns the population. If you run in the debugger, you can set breakpoints in either the client or the server and walk through the code.

## Add the Other Methods

Now that you've tested the round-trip connection, add the CensusServer Web Service's other two methods: GetAllPopulation() and GetTimeSeries(). Add definitions for the new functions in the class definition, as you did for GetStatePopulation(). Remember to mark

## VC++.NET • Retrieve a State's Population

```
[soap_method]
HRESULT GetStatePopulation(/* in */ BSTR stateID,
    /* out, retval */ long *population)
{
    USES_CONVERSION;
    // Convert the BSTR to a standard string:
    const string id = W2CA (stateID);
    map <string, long>::iterator i =
        CurrentPopulationStats.find (id);
    if (i != CurrentPopulationStats.end ())
        *population = i->second;
    else
        return S_FALSE;
    return S_OK;
}
```

**Listing 2** Implement the Web method that retrieves the current population for a single state. The code searches for the state ID in the data structures. If the code finds the state ID after searching for it in the data structures, the method returns the current population; otherwise, it returns an error.

```
<?xml version="1.0" ?>
- <serviceDescription name="CensusServerService"
    xmlns:s0="urn:CensusServerService" xmlns="urn:schemas-xmlsoap-
    org:sdl.2000-01-25" targetNamespace="">
- <schema targetNamespace="urn:CensusServerService"
    xmlns="http://www.w3.org/1999/XMLSchema">
- <element name="GetStatePopulation">
- <complexType>
- <all>
- <element name="stateID" type="string" />
- </all>
- </complexType>
- </element>
- <element name="GetStatePopulationResponse">
- <complexType>
- <all>
- <element name="return" type="int" />
- </all>
- </complexType>
- </element>
- </schema>
- <soap xmlns="urn:schemas-xmlsoap-org:soap-sdl-2000-01-25">
- <service>
- <requestResponse name="GetStatePopulation"
    soapAction="#GetStatePopulation">
- <request ref="s0:GetStatePopulation" />
- <response ref="s0:GetStatePopulationResponse" />
- </requestResponse>
- <addresses>
- <!-- apply to all interactions -->
- <address uri="http://localhost/CensusServer/
    CensusServer.dll?Handler=Default" />
- </addresses>
- </service>
- </soap>
</serviceDescription>
```

**Figure 2 View the Service Contract.** VS.NET creates a service "contract," or description, that lets client programs build proxies to communicate with your Web Service.

# Part of VSLive! Orlando

# SQL

## 2TheMax

Orlando • October 9-14

**Register Today!**  
[www.sql2themax.com/vsm](http://www.sql2themax.com/vsm)  
Call 800-848-5523

**Hard-Core Development Tools** for SQL Server 2000 are what you'll get at SQL2TheMax Orlando. From performance tuning to XML and e-commerce, SQL2TheMax helps you build today and tomorrow's solutions.

- ◆ FULL BONUS DAY ON .NET
- ◆ Dive Deep into T-SQL Coding and Stored Procedures
- ◆ Intensive Drill Downs with Microsoft Analysis Services
- ◆ Enhance Optimization and Performance Tuning
- ◆ E-Enable Your Database with XML, ADO.NET, ASP.NET

Call **800-848-5523** to Register

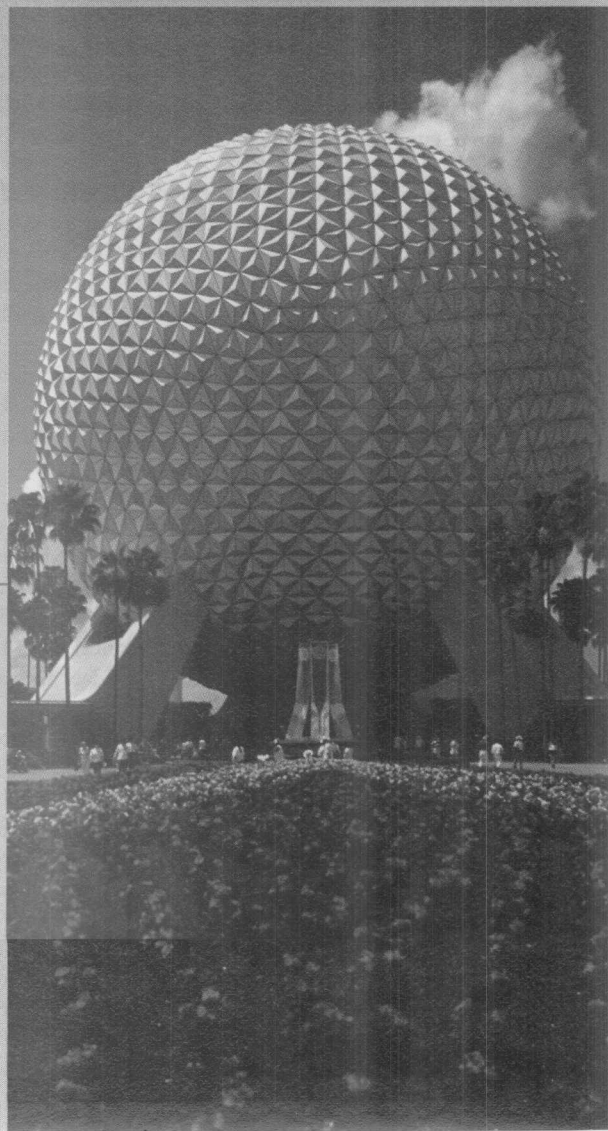
[www.sql2themax.com/vsm](http://www.sql2themax.com/vsm)

**FTP** FAWCETTE  
TECHNICAL  
PUBLICATIONS

**Visual Studio**  
Magazine

**Microsoft**

**msdn**



Microsoft, MS, MSDN, SQL Server, Visual Basic, Visual C++, Visual Studio, and Windows are registered trademarks of Microsoft Corporation. VBITS and Visual Studio Magazine are registered trademarks and Visual C++ Developer Conference, SQL2TheMax and SQLPro are trademarks of Fawcette Technical Publications, Inc. Visual Basic is used by Fawcette Technical Publications, Inc. Visual Studio is used by Fawcette Technical Publications, Inc. under license from Microsoft. All other trademarks are property of their respective owners.



## VC++.NET • Count Up All the States

```
[soap_method]
HRESULT CCensusServerService::GetAllPopulation
(/*[in,out] */ int *count,
/* [out,size_is(*count)] */ StatePopulation**
pStates)
{
    USES_CONVERSION;

    *count = CurrentPopulationStats.size ();

    *pStates = (struct StatePopulation*) malloc
        (sizeof (StatePopulation)*(*count));

    map<string, long>::iterator i =
        CurrentPopulationStats.begin ();
    int index=0;
    while (i != CurrentPopulationStats.end ())
    {
        (*pStates)[index].state = SysAllocString
            (A2W(i->first.c_str ()));
        (*pStates)[index].population = i->second;
        i++;
        index++;
    }
    return S_OK;
}
```

**Listing 3** This code implements GetAllPopulation(), the Web method that retrieves the current population for all states. This method allocates the storage for the array of nodes and populates each node with the necessary data. The count parameter returns the array size; the pStates parameter holds the array of nodes.

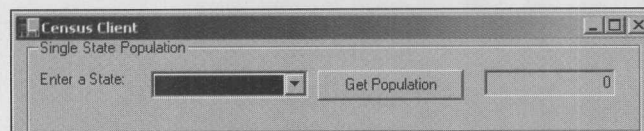
each method with the [soap\_method] attribute. The functions in the sample simply build the data structures to return and send them back (see Listing 3 for GetAllPopulation(); for GetTimeSeries(), download the code from the VSM Web site). Build and run the Web Service. Note the Service Definition Language (SDL) file has changed to include the new methods you just added.

Finally, add methods in the client application to test these methods. Update the Web reference code in the client project by right-clicking on the "localhost" node in the Solution Explorer and selecting Update Web Reference. This updates the Web reference proxy so it generates code for the new version.

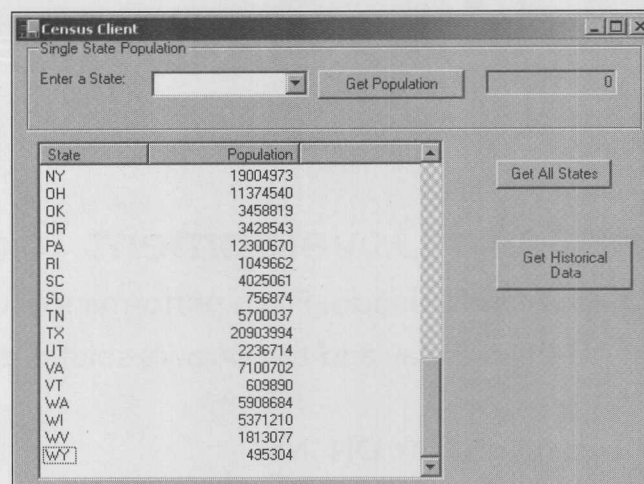
Next, add a ListView control and two buttons to test the new methods (see Figure 4). The code to test these two new methods is pretty straightforward. Simply call the service, iterate through the results, and add them to the listview (see Listing 3). Clicking on either the Get All States or Get Historical Data button fills the listview with all the data. The listview's data and column headers change depending on which button you push.

Notice I spent little time discussing the framework necessary to create a Web Service using ATL Server—VS.NET's Web Services framework does all the grunt work common to all Web Services. The framework and code attributes provide all the necessary code to turn any class into a Web Service. Simply create the class you want with the public methods to implement your interface.

You should also consider using C# to build your Web Services. I've built a Web Service using both C# and VC++.NET. The C# version provides higher-level tools, and you can avoid even more of the COM plumbing. The CensusServer Web Service doesn't require extensive COM programming, but you'll need to think about that when you



**Figure 3 Consume the Web Service.** This simple C# application, Census Client, can access the CensusServer Web Service and use its results. The user selects a state from the combo box and hits the Get Population button to display that state's population.



**Figure 4 Ready, Get Sets, Go!** The same C# client application can retrieve sets of data from the Web Service. More sophisticated clients could provide sorting and graphing capabilities for data analysis.

build more complex Web Services. On the VC++.NET side, ATL Server lets you use some of the more powerful tools in the C++ standard library. If you were to develop this simple Web Service more fully, you'd add searching and sorting algorithms to manipulate the data. Also, you'd probably use more extensive numerical analysis tools to find patterns and trends in the data. You can write these algorithms more easily using C++, and consequently, ATL Server. **VSM**

**Bill Wagner** is a founder of SRT Solutions ([www.srtsolutions.com](http://www.srtsolutions.com)). He has 15 years of programming experience and has taught several classes using Microsoft development tools. He is writing *C# Core Language Little Black Book* (The Coriolis Group, due out in Fall 2001). E-mail Bill at [wwagner@srtsolutions.com](mailto:wwagner@srtsolutions.com).



## Go Online

Use these DevX Locator+ codes at [www.vbpi.com](http://www.vbpi.com) or [www.vcdj.com](http://www.vcdj.com) to go directly to these related resources.

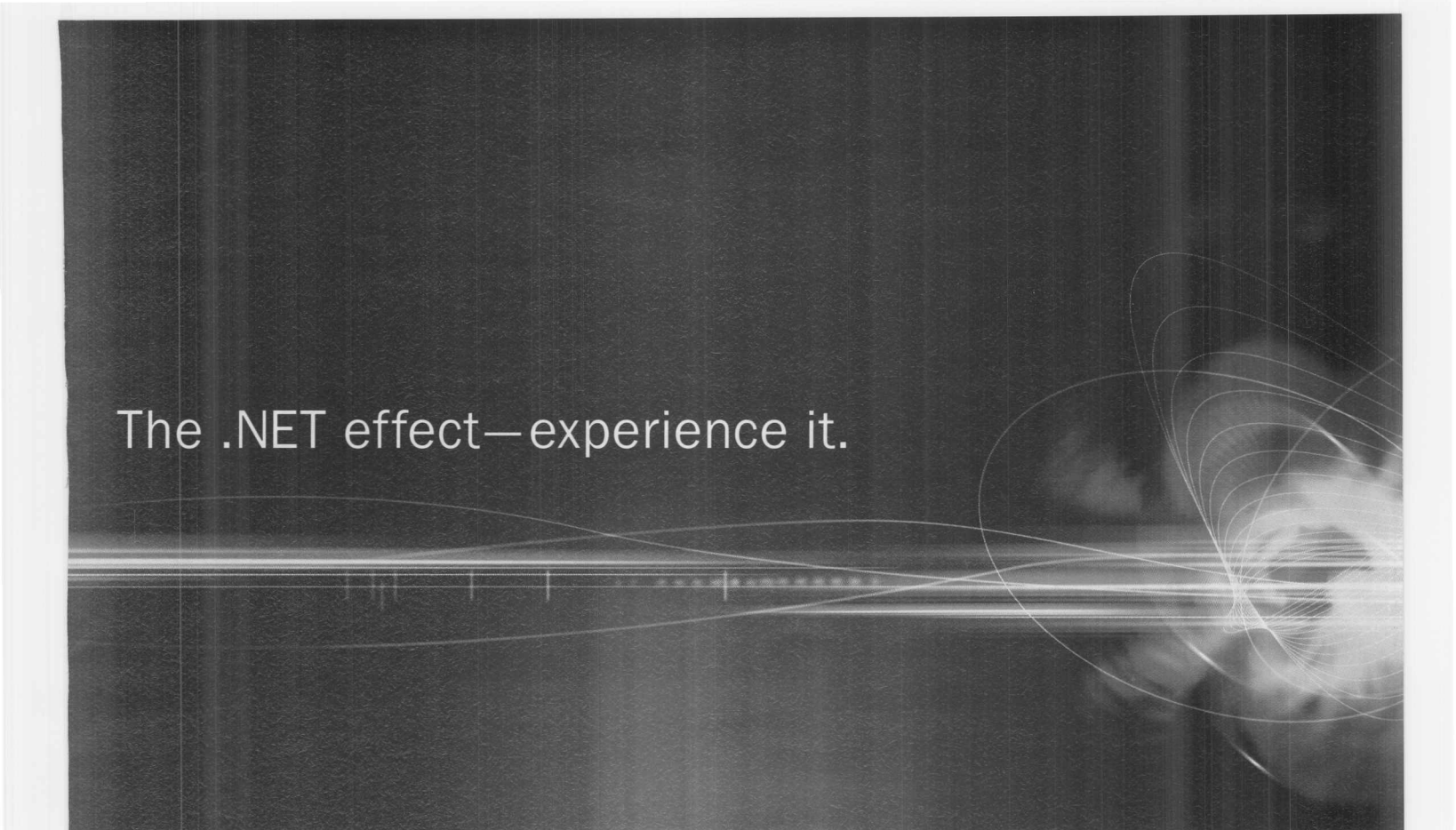
**VS0110** Download all the code for this issue of VSM.

**VS0110BB** Download the code for this article separately. This article's code includes a Web Service built using C++ and ATL Server, that same Web Service built using C#, and a C# client that uses the Web Service.

**VS0110BB\_T** Read this article online. DevX Premier Club membership is required.

**Want to subscribe to the Premier Club?** Go to [www.devx.com](http://www.devx.com).





# The .NET effect—experience it.

## Microsoft Professional Developers Conference 2001

October 22–26, 2001

Los Angeles Convention Center

[www.microsoft.com/events/pdc](http://www.microsoft.com/events/pdc)

## Discover what's next for .NET.

You don't want to miss this turning point. PDC 2001 is the only place you'll see the future firsthand:

- Be the first on your block to experience the latest XML Web services technologies, including "HailStorm" (code name), the .NET Framework, and more.
- Experience the evolution as **Bill Gates**, Chairman and Chief Software Architect, updates you on Microsoft's strategy and reveals the latest advances for .NET.
- Learn how to leverage these new technologies from key Microsoft architects and executives.
- Get the exclusive inside view of breakthrough tools and technologies to shape the next generation of the Internet.

The net gain? You'll be out front with the vision, insights, and code everyone wants. The net effect? You're on course to pioneer capabilities that deliver incredible new user experiences.

**Microsoft® .NET is leading the industry toward a new generation of software services, and you can be at the forefront of it all—but only if you register for PDC 2001 now: [www.microsoft.com/events/pdc](http://www.microsoft.com/events/pdc)**

## Compress That Data!

Add royalty-free data compression to **all** of your applications.

**DynaZIP™** Data Compression Tools

**Active Delivery™** Self-extracting Zip Tools

- Widest range of language support.
- ActiveX/DLL/VCL interfaces, more.
- Most reliable, *millions* in use daily.
- Fully supports Active Server Pages.
- New GZip/Tar Version Available.

Download your free eval copies today!

 inner media 800-962-2949 (USA)  
603-465-3216  
[www.innermedia.com](http://www.innermedia.com)

## RichPayments.NET™

New ePayment Web Service!  
Process Payments on any Device, at any  
Time and any Place.

- Interoperable – built on SOAP & XML.
- Supports credit cards, debit cards and checks.
- Works in traditional and eCommerce applications via easy to integrate DLL.
- Supports real-time & recurring billing.
- Add value to your applications or use to bill for your own web services.
- Free SDK with low monthly service fees.

RichSolutions, Inc.  
Phone (425) 836-4251  
<http://www.richsolutions.com>

NEW!  
Version 2.8

## Visual UML

Object modeling using the Unified Modeling Language

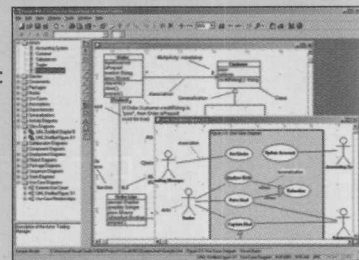
Visual UML™ is an affordable, easy-to-use, yet powerful and full-featured object-modeling tool that supports all OMG 1.3 UML (Unified Modeling Language) diagram types: Use Case, Package, Class, Object, State, Collaboration, Activity, Deployment, Component and Sequence diagrams.

**Features:** Integrated model/data dictionary; project explorer; link objects to diagrams, files, URLs; group editing; use case editor; print/export preview; import/export/merge models & diagrams; XML interface (with DTD); publish models & diagrams to HTML; export diagrams to WMF, BMP, JPEG, TIFF, EMF, etc; stereotypes; tagged values; **data modeling; database schema generation & reverse-engineering; C++, C# & Java code generation; reverse-engineer C++ code (using CC-RIDER & interface); VBScript & JScript 5.5; automation interface, and more!**  
**Visual UML Plus Edition** adds: integrated support for Visual Basic for Applications (VBA) 6.2.  
**Visual UML for Visual Basic** adds: integrated code generation and reverse-engineering of Visual Basic projects. Can also be run as an add-in to Visual Basic.  
**Visual UML for Visual FoxPro** adds: integrated round-trip code generation and reverse-engineering of Visual FoxPro projects.

**Also available: CC-RIDER:** C++ source code analysis, documentation and visualization. Reverse-engineering of C++ code. Interface to Visual UML.

**Pricing: Starts at only \$495 (Visual UML Standard Edition)**

Includes the best-selling, award-winning book: **UML Distilled**, by Martin Fowler  
Download a **FREE** trial/evaluation version of Visual UML from our web site.



Visual Object Modelers • [www.visualuml.com](http://www.visualuml.com)

Tel: 1-800-900-1902 or (303) 448-8835 • Fax: (303) 415-2500

Email: [info@visualobject.com](mailto:info@visualobject.com)

## Bar Code ActiveX and DLLs



"Bar codes  
look fantastic...  
TAL has the  
market on  
this one."

— VB Programmer's  
Journal

Easily add professional bar coding capabilities to Windows applications in minutes. High-quality device-independent bar codes. Not fonts! Not bitmaps! Perfect for VB, C++, Access, Excel, Word and Internet Explorer. Supports data binding. Prints all major bar code types — including 2-D.

**TALtech**

Free demos at [www.taltech.com](http://www.taltech.com)

800-722-6004

## Accounting Source Code

Integral Accounting Basic VB is a full featured Accounting System that includes OE, Inventory, AP, AR, GL, Payroll, and Source Code written in VB 6.0. Printed and electronic documentation are also included. To download a demo, please visit [www.stfb.com](http://www.stfb.com)

(Access & SQL Versions also available)

**STFB Inc.**  
**PH: (954) 986-8917**  
**<http://www.stfb.com>**

## 8 Cool Controls!

- TE Edit Control Advanced RTF control
- HTML Viewer/Editor Add-on for TE
- SpellTime DLL and dictionary
- RTF Code Generator
- ReportEase Plus report writer engine
- Form Plus
- Rich Text Grid
- ChartPro

**Sub Systems, Inc.**  
**TELE: (978)352-9020 FAX: (978)352-9019**  
**Demos: [www.subsystems.com](http://www.subsystems.com)**



## Job Scheduling

SmartBatch32 offers fully integrated mainframe-like job batch scheduling at a fraction of the cost. SmartBatch32 enables full control of jobs with dependencies, sequencing, flexible scheduling, and notification. Schedule across multiple computers with the Enterprise Edition. Free eval downloads:

**OnLine Toolworks Corporation**  
<http://www.OnLineToolWorks.com>  
**eMail: sales@OnLineToolWorks.com**

## eCommerce & Accounting

Looking for a maintenance-free shopping mall or a fully customizable accounting software? We have a full accounting package (written in VB) designed for midmarket and online store that you can maintain through our accounting package. SQL Server/Oracle/Access/Crystal Report/360 reports. Try our functional demo today.

**DynasticSoft Corp**  
949.263.5400  
[www.DynasticSoft.com](http://www.DynasticSoft.com)

*Are You Tired of Paying Royalties?*

## SKYLINE TOOLS IMAGING PRODUCTS

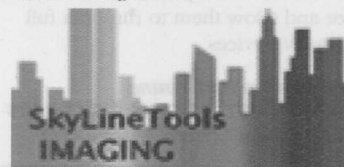
### Royalty Free Products

- **CORPORATE SUITE ACTIVE X**  
*Document and Color Imaging*  
Color and Document Imaging Package With Source
- **IMAGELIB BARCODE SUITE**  
*Barcode Reader*
- **IMAGELIB OMR TOOLKIT**  
*Optical Mark Recognition*
- **VIDEOLIB OCX**  
*VideoCapture*

- **DOC-TO-NET**  
*Internet Imaging*  
Send documents over Internet

### Free Trial Downloads:

[www.imagelib.com](http://www.imagelib.com)  
Phone: (818) 346 4200  
email: sales@imagelib.com



## Visual Basic Compiler and Syntax Services

Specializing in VB compiler related technology services. Performing automated legacy source code conversions between VB and C, C#, Java, Delphi and others. Services for creating specialized VB parsers. OEM implementations of VB language on non-Windows platforms. Large project orientation.

**Softworks Limited**  
[sales@SoftworksLtd.com](mailto:sales@SoftworksLtd.com)  
[www.SoftworksLtd.com](http://www.SoftworksLtd.com)

## WAP Applications

Building wireless and WAP applications—THE SIMPLE WAY! Design user-interface, database access and flow control for your wireless application VISUALLY. Applications built are compatible with different wireless devices including WAP phones and RIM's Blackberries. Download our application builder and start building wireless applications TODAY!

**Xybo Systems Inc.**  
Ph. (905) 943-7482 Fax (905) 470-8975  
Email: [contact@xybo.com](mailto:contact@xybo.com) Web: [www.xybo.com](http://www.xybo.com)

Thinking about  
**IT Certification?**  
Listen up.  
[There's going to be a test later.]

Transcender's team of development experts leads the way to certification with realistic and accurate exam preparation tools for **Microsoft®**, **A+®**, **Network+™**, **i-Net+™** and **Cisco®** Certification.

**TranscenderFlash™** quiz software drills you on your knowledge of the fundamentals while **TranscenderCert™** exam simulations test your readiness for the real thing.

**TranscenderCert** features include **customized exams**, **randomized exams**, **simulation questions**, a **score report history**, and **computer adaptive testing**. With Transcender, you can be absolutely sure when you are ready. Be confident you'll pass with Transcender's Money-Back-If-You-Don't-Pass Guarantee.\*

Visit [www.transcender.com/vbpj](http://www.transcender.com/vbpj) and download FREE demos.

Order and download Transcender products at  
[www.transcender.com/vbpj](http://www.transcender.com/vbpj),  
call 615.726.8779 (8 a.m. - 6 p.m., CT, M-F),  
fax 615.726.8884, or mail to 565 Marriott Drive,  
Suite 300, Nashville, TN 37214.

[www.transcender.com/vbpj](http://www.transcender.com/vbpj)

**Transcender®**  
AN INFORMATION HOLDINGS INC. COMPANY



**Certification. Guaranteed.\***

Copyright © 2001 Transcender LLC. All rights reserved. Transcender, TranscenderFlash, and TranscenderCert are either registered trademarks or trademarks of Transcender LLC. Microsoft is a registered trademark of Microsoft Corporation. A+, i-Net+, and Network+ are either registered trademarks or trademarks of CompTIA. Cisco, CCNP, and CCDP are registered trademarks of Cisco Systems, Inc. Multi-user licenses are available. \*Call or see our Web site for details on the Money-Back-If-You-Don't-Pass Guarantee.





## ImageKit5 and ResizeKit

The **ImageKit5** adds image-processing capabilities to your applications. Includes Display, Effect, File, Print, Scan, Slideshow, and Thumbnail Controls with over 110 functions.

The **ResizeKit** automatically makes your applications size and resolution independent. No coding necessary. Contains both ActiveX and VCL controls.

**NEWTONE Corporation**  
<http://www.newtone.co.jp>

## Report Writer OCX

- End-user Ad Hoc report designer
- Optimized for MS Access databases
- No coding required
- Customizable via optional code
- Includes integrated Query Builder
- WYSIWYG print preview
- Save report files in XML format
- Royalty free, lifetime free support

**Application Design Consultants**  
**Phone: (877) 343 - 0229**  
**Demo: www.ApDC.com**

## Reach the Visual Studio Community—In Person!

**Exhibitor and Sponsorship Opportunities at**

**VSLive!**  
 Visual Studio Developer Conference

**VBITS**  
 VISUAL BASIC INSIDERS' TECHNICAL SUMMIT

Please Contact: Tim G. Smith  
**650.688.2469**  
[tsmith@fawcette.com](mailto:tsmith@fawcette.com)

[www.vslive.com](http://www.vslive.com)  
[www.vbits.net](http://www.vbits.net)

## ReSize

ReSize gives your applications resolution independence. ReSize alters the sizes, font sizes, and positions of controls on Visual Basic forms as they are sized without requiring you to write code. ReSize will give your applications a more professional appearance and allow them to run on a full range of display devices.

**Larcom and Young**  
**TELE: (810)771-9597 FAX: (810)771-9564**  
**Free demo: <http://www.lyoung.com>**

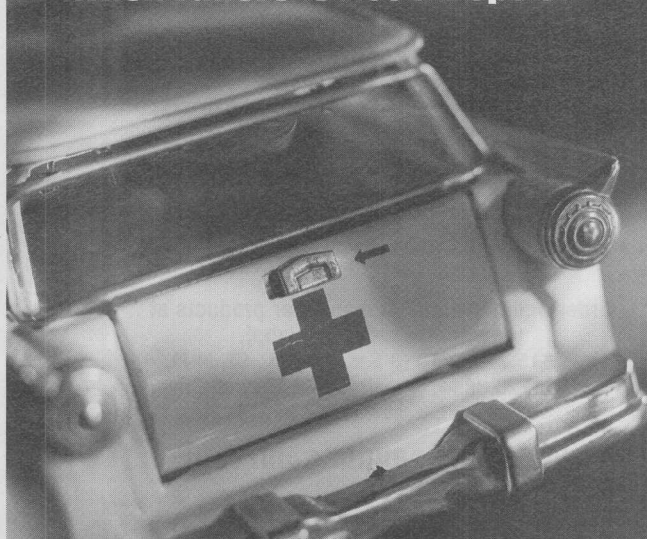
**Exhibitor and Sponsorship Opportunities at**

**XML**  
 MAGAZINE'S  
**Tech Summit**

**Washington, D.C., Oct. 22-24 2001**

Call: Tim G. Smith  
**650.688.2469 • [tsmith@fawcette.com](mailto:tsmith@fawcette.com)**  
[www.xmltechsummit.com](http://www.xmltechsummit.com)

**Sax Comm Objects provides mission-critical communications when failure is not an option.**



Every day, thousands of applications around the world rely on Sax Comm Objects for robust, fail-safe modem communications. Sax Comm Objects is MSComm and PDQComm compatible, but is faster, easier to use, and more robust. Download your free trial today.

**1-800-645-3729**  
**[www.SaxComm.com](http://www.SaxComm.com)**  
 Use value code 3577 to receive a 10% discount.

## The *Fastest* Database Engine

Discover **CodeBase**, the **fastest** database engine on the market that's also fully **xBASE compatible**. Query a million-record table in 0.34 seconds, or read 300,000 records in just 0.59 seconds. It's lightning quick!

Product Inventory		Product	
Product:	Widget Bearings		
Product Code:	BDS-32450		
Description:	Widget Bearings are mainly used in the production of fashions and dancet computers.		
Product Code:	Product:	Units Stocked:	100
BDS-32450	Widget Bearings	Cost / Unit:	\$48.00
ACE-32451	Buffer Wrench	List Price:	\$59.99
BDS-32452	Dumet Joints	Stock Value:	\$4,800.00
CEE-34654	Footbar	Units Ordered:	3000
DSC-46454	Clismo	Total Cost:	\$147,000.00
REE-47653	Widget Bearings	Date Ordered:	March 14 / 1997

### ADO/ODBC Support!

This application was created without a single line of source code! Use native and 3rd party controls with our new ADO/OLE-DB and ODBC support. Creating database programs has never been easier!

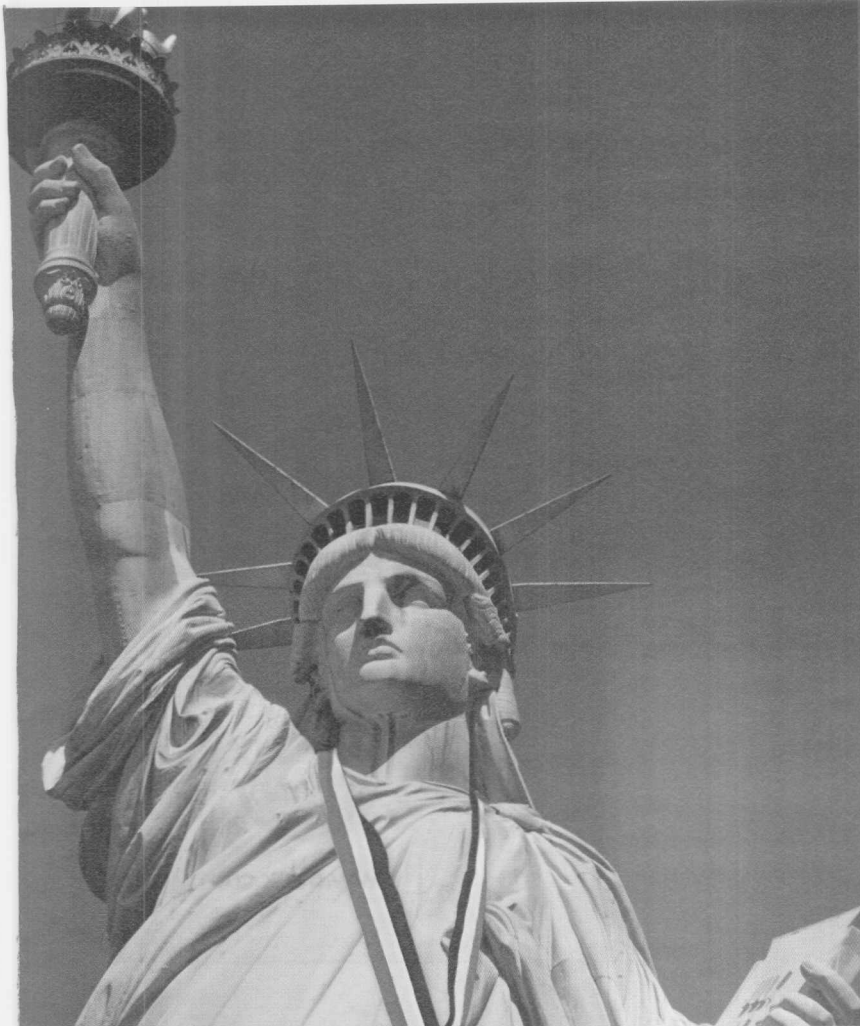
- ✓ Multi-user file compatible with FoxPro, Clipper and dBASE
- ✓ Client/Server, multi-user & single user support included
- ✓ Small DLL loads quickly, simply and use few resources
- ✓ 100% Portable: Windows, Pocket PC etc. 32 & 64 bit
- ✓ Supports VB, C, C++, Delphi, Java etc.
- ✓ Reader's Choice Award for 5 Years
- ✓ Royalty Free Distribution

**FREE 30 Day Test Drive!**

**SEQUITER**  
 SOFTWARE INC.

[www.sequiter.com](http://www.sequiter.com) - [info@sequiter.com](mailto:info@sequiter.com)

Phone: (780) 437-2410  
 Fax: (780) 436-2999



The Component Vendor Consortium (CVC) is a non-profit organization committed to ensuring developers have access to well-written, thoroughly tested components that can be implemented without difficulty. To help you choose components that meet exacting quality standards, the Component Vendor Consortium has initiated the CVC Quality Certified logo program. This program subjects components to a rigorous series of tests, overseen by Compuware Professional Services, an independent testing service. To qualify for logo certification, components must achieve at least 80% code coverage and must be proven to contain no unexplained memory corruption errors, be free of memory leaks, and have no unhandled errors or exceptions. When you select a component that carries the CVC Quality Certified logo, you can have a feeling of confidence, knowing you're using tools designed to keep you developing, not trouble shooting.

## Look for the CVC Quality Seal.

For more information on the CVC quality certification process, including details on the testing process and procedure, visit the CVC web site at:

<http://www.components.org/quality.asp>

The CVC is sponsored by:

Microsoft Corp., Fawcette Technical Publications, Advisor Media,  
Software Development Times and CMP Software Development Media Group

**Microsoft®**

**FTP** FAWCETTE  
TECHNICAL  
PUBLICATIONS

**ADVISOR  
MEDIA**

**SD Times**  
The Industry Standard in Software Development Strategies

**CMP**  
Software  
Development  
Media  
Group

# Quality Deserves Recognition



The general membership of the CVC consists of companies that develop and market commercial software components and/or development tool products. CVC members include leading vendors in the component marketplace such as

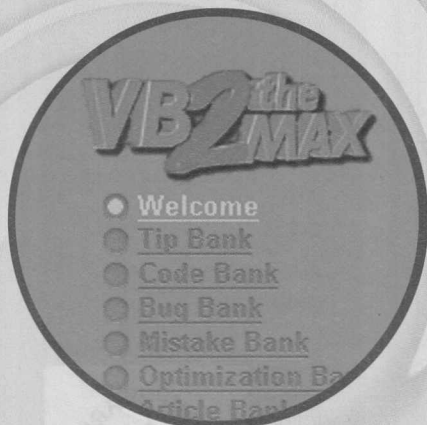
Advantageware,  
Bennet-Tec Information Systems, Inc.,  
BestofWare,  
Bunka Orient Corporation,  
ComponentOne,  
Compuware NuMega,  
CORDA Technologies,  
DART Communications,  
Data Dynamics, Ltd.,  
DBI Technologies,  
Desaware, Inc.,  
Distinct Corporation,  
FarPoint Technologies,  
FMS, Inc.,  
Infragistics, Inc.,  
LEAD Technologies, Inc.,  
Sax Software,  
SoftArtisans,  
VisualSoft Technologies Limited.



Access Expert Visual Basic® Programming Information—Fast.

## VB2TheMax—The Knowledge Repository for VB Developers

Go to the VB2TheMax area on the DevX VBZone for quick access to selected routines, tips, articles, and sample chapters on VB5, VB6, VB.NET, Windows, SQL, ASP, and more!



### Updated Weekly:

- 800+ tips, routines and bug reports
- Original articles and sample chapters from the best VB books
  - 6000+ selected Microsoft Knowledge Base articles
  - Index of over 900 VB articles

Get Your **FREE** newsletter for even more, including the new VB.NET Watcher column.

**Sign up today!**

[www.vb2themax.com](http://www.vb2themax.com)



[www.devx.com](http://www.devx.com)

DevX.com, Inc. • 913 Emerson Street, Palo Alto, CA 94301 • 650.566.2000

DevX is a registered trademark of DevX.com, Inc. VB2theMax is a trademark of Francesco Balena. Visual Basic and Windows are registered trademarks of Microsoft Corporation.

## DevX Delivers!

Get the Development Information You Need—Delivered to Your Inbox.

DevX newsletters deliver the articles and information you need to keep your projects moving. Choose from over 20 development topics.

**Sign up today!**

[www.devx.com/newsletters](http://www.devx.com/newsletters)

### Free Weekly VB Newsletters

- ☒ VB Update
- ☒ VB2TheMax
- ☒ Weekly DevX HotLinks
- ☒ .NET Update
- ☒ ASP Update
- ☒ Database Development Update
- ☒ Security Update
- ☒ MarketPlace Specials



[www.devx.com](http://www.devx.com)

DevX.com, Inc. • 913 Emerson Street, Palo Alto, CA 94301 • 650.566.2000

DevX is a registered trademark of DevX.com, Inc. All other names are the property of their respective owners.



# Get Answers to Your Toughest Questions from Qualified Experts

Have a mission-critical  
question you need  
answered fast?

DevX Knowledge Xchange delivers the fast answers  
you need to keep your project moving by soliciting  
user-qualified development professionals who *compete*  
to give you the best solution.

- Search over 2 million timely, real-world IT development solutions
- Power search our vast library of answered questions
- Choose the best solution among qualified answers
- Share your knowledge with your peers
- Easily manage your personalized content!

As low as  
**\$16**  
per month

With the purchase  
of a yearly  
membership




## Sign up now!

Go to  
**DevX Knowledge Xchange at:**

[www.devx.com/knowledgexchange](http://www.devx.com/knowledgexchange)

  
**www.devx.com**



the world's intellectual capital exchange  
post a project, save \$500 HelloBrain.com

**The Development Exchange**

Zones:  
Application Server  
ASP  
C++  
CareerLink  
DHTML  
Enterprise  
Java  
JavaBeans  
Linux  
NetMeeting  
Security  
SQL  
XML  
Visual Basic  
XML

Magazines:  
Enterprise Dev  
Exchange & Outlook  
Getting Started  
Java Pro  
VB Pro  
VFP  
VFP  
Web Builder  
XML

MarketPlace:  
Bookstore  
Software  
Subscriptions  
Premier Club

Conferences:  
Builder.com Live!  
DevCon

**SEARCH**

Search for:  Libraries:  Search

**Today's News**

- Check Out the New DevX Security Zone
- The Net Effect on Politics
- ASP Partners Give Boost to Provider Market
- Eazel Aims to Put a New Face on Desktop Computing
- Is the Federal Fox Really Guarding the Online Privacy Henhouse?
- More News from InfoWorld Magazine

**Highlights**

<p><b>FEATURE ARTICLES</b></p> <ul style="list-style-type: none"> <li>Improve MTS and COM+ Performance —from the September VBP</li> <li>Scaling the Enterprise: IPPlanet Application Server —from the September Java Pro</li> <li>When Time Is of the Essence —from the July Enterprise Development</li> </ul>	<p><b>HOTLINKS NEWS</b></p> <ul style="list-style-type: none"> <li>Eight (More) Essential VS.NET Answers</li> <li>Previous HotLinks</li> </ul> <p><b>DOWNLOADS</b></p> <ul style="list-style-type: none"> <li>Project.Backup 2.0</li> <li>DeZign for databases 2.2</li> <li>PowerTCP Server Tool 2.0</li> <li>Simplicity for Java 1.2.1</li> </ul>	<p><b>TIPS</b></p> <ul style="list-style-type: none"> <li>VB - Change Dates With Plus and Minus Keys</li> <li>ENT - SQL Stored Procedure to Run Single Command on All Tables</li> <li>C++ - Rules of Thumb for Software Internationalization</li> <li>JAVA - Increase Code Reusability</li> </ul>
--	--	---

**DevX MarketPlace** **Specials!** **Current Top Sellers**

Slash Your: The VBETS New York Ten Callers **Software!** VideoSoft VSPlexGrid Pro

Sponsored by:

Click here for details.

Microsoft

\$300

**Join Premier Club FREE for 30 days**

click here

Vote

All hardware and software being equal, which OS would you prefer?

☐ UNIX/Linux

☐ Windows 2000/98/95/NT

☐ Macintosh

☐ BeOS

☐ Other

Vote

**Upgrade NOW**

## Premier Club Delivers:

- Savings on software
- Access to the DevX development article and code archive
- Live member-only events

Join the Premier Club for \$3.95 per month and your first 30 days are free!

From the DevX home page click on JOIN

# Need Development Know-How?

## Your search stops at DevX.

Get the unbiased information and tools you need NOW from the leading information resource for IT professionals developing business applications. We connect developers with objective, technical information and tools to get the job done—code samples, technical articles, discussions, learning services, and other resources aimed at helping you solve your toughest development challenges.

To learn more, visit:

**www.devx.com**

DevX is a trademark of DevX.com, Inc.

- Tap into 1000's of programming tips
- Ask the experts
- Browse tons of content and code from top IT publications
- Join live developer forums
- Access over 2,000 product reviews
- Purchase and download leading development tools
- **Join Premier Club** for exclusive benefits



developmentexchange

# Advertiser Index

## Fast, Free Product Information

Advertiser Telephone; Web site	Page	Advertiser	Page	Advertiser	Page
<b>AppDev</b> 800-578-2062; www.appdev.com	81	<b>Gimpel Software</b> www.gimpel.com	69	<b>Rainbow Technologies</b> www.rainbow.com	77
<b>AppForge</b> www.appforge.com	37	<b>HiT Software</b> 408-345-4001; www.hitsw.com	32	<b>Rational Software</b> www.rational.com	6
<b>Apress</b> www.apress.com	20	<b>IBM</b> www.ibm.com	73	<b>Sax Software</b> www.saxsoft.com	106
<b>ComponentOne</b> 800-858-2739; www.componentone.com	4, 31, 47, 67, 93	<b>Infragistics</b> 800-231-8588; www.infragistics.com	1, C2, C3, C4	<b>Sequiter Software</b> 403-437-2410; www.sequiter.com	68, 106
<b>Component Vendor Consortium</b> 800-637-4699; www.components.org	107	<b>INNERHOST</b> www.innerhost.com	22	<b>Softel vdm Inc.</b> www.softelvdm.com	34
<b>Compuware</b> www.compuware.com	89	<b>InstallShield</b> www.installshield.com	12	<b>SoftWIRE Technology</b> www.softwiretechnology.com	24
<b>Data Dynamics</b> 614-895-3142; www.datadynamics.com	35	<b>InterSystems Corp.</b> www.intersystems.com	16	<b>SourceGear</b> www.sourcegear.com	30
<b>dbi technologies</b> www.dbi-tech.com	10	<b>Kenonic Controls Ltd.</b> 403-258-6200; www.crypkey.com	49	<b>SQL 2 The Max</b> 800-848-5523; www.sql2themax	101
<b>Desaware</b> 408-377-4770; www.desaware.com	15	<b>Keystone Learning Systems</b> 800-349-5761; www.klscorp.com/vpj	38, 39	<b>VCDC Orlando</b> 800-848-5523; www.vcdc.com	50
<b>DevX</b> 888-963-3389; www.devx.com	110	<b>LEAD Technologies</b> 800-637-1842; www.leadtools.com	45	<b>VSLive! Orlando</b> 800-848-5523; www.vslive.com	97
<b>DevX Get Answers</b> 888-963-3389; www.devx.com/knowledgeexchange	109	<b>Mabry</b> 800-99-MABRY; www.mabry.com	33	<b>Wise Solutions</b> 800-554-8565; www.wisesolutions.com	63
<b>DevX VB-2-The-Max</b> www.vb2themax.com	108	<b>Microsoft</b> www.microsoft.com	2, 3, 18, 19, 26, 27, 103		
<b>DevX Newsletters</b> 888-963-3389; www.devx.com	108	<b>Microsoft Web Services Contest</b> www.webservicescontest.com	36		
<b>dtSearch</b> 800-IT-FINDS; www.dtsearch.com	60	<b>.NET Magazine Subscriptions</b> 800-848-5523; www.fawcette.com/dotnet	48		
<b>FarPoint Technologies</b> 800-645-5913; www.fpoint.com	82, 83	<b>/n Software</b> 919-544-7770; www.nsoftware.com	88		
<b>FMS Inc.</b> 888-220-6234 ext. 222; www.fmsinc.com	87	<b>Osborne McGraw-Hill</b> www.osborne.com	55		
<b>Gigasoft</b> www.gigasoft.com	61	<b>Pegasus</b> 800-875-7009; www.pegasustools.com	43		
		<b>Programmer's Paradise</b> 800-445-7899; www.pparadise.com	8, 9		
				<b>Product Showcase</b>	104-106

► To get immediate details on products advertised in this issue, you can simply log onto our online services at <http://infoasap.devx.com>.

► This will provide a list of direct links to this month's advertisers' Web sites.

**Advertising Sales:** Karin Becker, Group Advertising Director 650-833-7168 • Brigit Chennells, Regional Account Manager (Southwest & Southeast) 949-369-5138 • Mark Romasco, Regional Account Manager (Northeast) 978-779-8989 • Tara Stafford, Regional Account Manager (Northwest) 949-369-5232 • David Seymour, Inside Sales Manager 650-833-7148 • Fina Pita, Production Coordinator 650-833-7140

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions. Sales Department: 209 Hamilton Avenue • Palo Alto, California, USA, 94301-2500 • 650-833-7100 • 800-848-5523



# Put Rich and Thin Together

**N**etscape introduced the world's first commercial browser in 1994. At that time, local area networks (LANs) were really taking off and businesses were connecting their offices into wide area networks (WANs), sometimes using the Internet as an established backbone. The infrastructure existed to allow users in any office to connect to the main office at the network level, but applications still couldn't communicate easily over this infrastructure.

That's when IT professionals realized: Why not build browser-based thin clients? Ever since, there has been a tremendous shift in application development. Browser-based intranet applications are now commonplace, but is this a good thing? That depends on your needs.

As an IT professional, you probably think thin clients are great because you don't have to deploy anything to the desktop. After all, who wants to deal with installation issues, overwritten DLLs, and a user out in the middle of nowhere whose machine stopped working after he installed the latest version of your Windows app? You probably think thin clients are great because they eliminate all this hassle and save deployment costs. You're right.

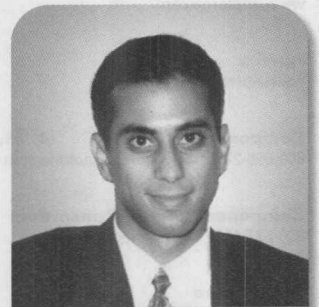
But talk to a business end user, and you might hear a different story. This user probably likes the anywhere, anytime accessibility of a Web client, but dislikes the clunky user interface. It's no secret: Web UIs are way behind Windows UIs. Developers try to overcome this limitation constantly. Many Web applications try to offer a rich UI by leveraging IE DHTML features. Others make heavy use of ActiveX controls to enhance the user's experience. Unfortunately, such measures improve the UI only slightly, while taking away some of the deployment benefits. Deploying an ActiveX control to the desktop can be every bit as problematic as deploying a full-fledged Windows application. In fact, you can't deploy any ActiveX controls if the organization's security policies are sufficiently strict.

There's more to the problem than a poor UI. Web applications are also difficult to build and can grow into unmanageable beasts. This is due largely to the limited capabilities of HTML and scripting, which is especially true if you try to make your application browser-independent. Ultimately, developers spend more time building a Web application compared to a Windows application, with the end result being less than satisfactory for the business end user. Clearly something is wrong here.

With today's technologies, you can build rich Windows applications that are easy to deploy, leveraging the Internet or intranet for communication with the server. You can also package and deploy Visual Basic applications to the desktop over the

Web. However, the issues of applications stepping on each other's files remain a problem. Windows XP offers a solution to deployment issues using manifest-based, side-by-side deployment. You can find a more complete set of solutions in Windows Forms, remoting, and Web Services—all part of .NET. You can also build a rich client with Windows Forms and VB.NET, then deploy it to the desktop with no fear of stepping on another application's assemblies. This client can use .NET remoting to communicate with your server components over HTTP and through firewalls. The client calls the server components directly, so there's no need for a layer of HTML forms and Active Server Pages to mediate this interaction. Finally, the client can also leverage Web Services to communicate with server applications running on any platform and written in any language.

Deploying the .NET runtime to every desktop is the next big hurdle. If you think of the .NET runtime as equivalent to VB6's runtime, then deploying a .NET app is no different than deploying a VB6 ActiveX control. Speaking practically, however, the .NET runtime is much larger than VB6's runtime. There's also the small matter that it hasn't been released yet, so there aren't many applications that need this runtime at the moment. That will change as people realize that installing .NET on every desktop is like installing a browser on every desktop: You install a relatively fat client or runtime, then leverage that with thin application clients. With .NET, those thin clients can actually be rich Windows apps that are easy to develop and deploy. **VSM**



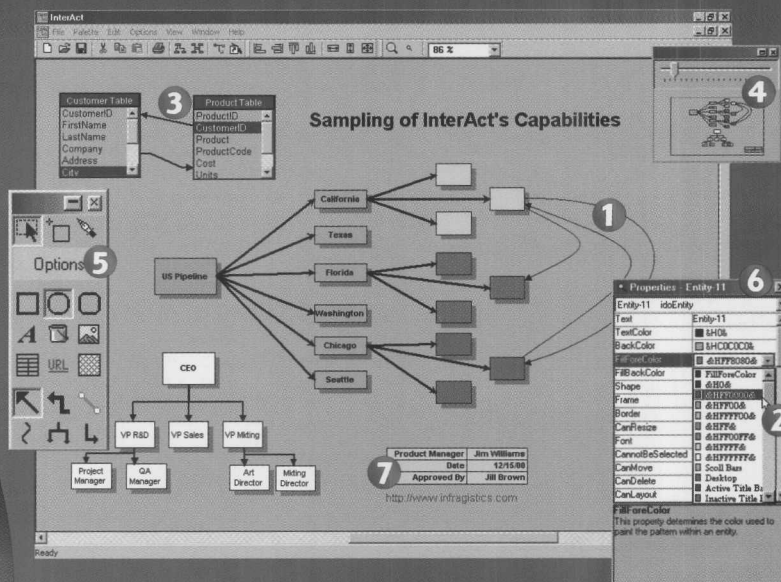
**Yasser Shohoud**

Agree? Disagree? State your opinion on the Talk to the Editors discussion area on The Development Exchange (<http://news.devx.com>).

The opinions expressed in this editorial are those of the author and not necessarily the opinions of VSM.

---

**Yasser Shohoud** has been a software developer for more than 12 years. He writes regularly for *Visual Studio Magazine* and *XML Magazine*. Yasser is an independent consultant specializing in building Web Services with VB and is authoring the book, *Building Web Services with Visual Basic* (Addison-Wesley). Reach Yasser at [www.vbws.com](http://www.vbws.com).



# Visualize This.

Infragistics - formed by the merger of Sheridan and ProtoView

## InterAct™ 4.0

Advanced data visualization features, a powerful programming model and ease-of-use make InterAct the world's leading COM-based **diagramming component**.

With InterAct's seamless integration into Visual Studio tools, professional developers can quickly create sophisticated, robust diagram-based applications using our pre-built or customizable objects. Take advantage of our Process Automation Engine, Layout Manager, Overview Window, printing/print preview and much more, in just a few mouse clicks.

- **NEW** Illustrate relationships clearly with List Entities. Programmatically or visually build lists of information and link items in lists between List Entity objects.
- **NEW** Get/set the properties of InterAct, the entities and the relations during run-time, with the included PropertyBrowser.
- Create custom entities easily, at design-time or run-time, including text entities for labeling diagrams, and tables in entities for displaying legends and grids of data.
- Place images (bitmaps or metafiles) in entities, with transparency option and fill patterns.
- Control font, size and color specifications for all entities, relations and diagram attributes.
- Provide only the specific objects needed by your user in the custom Tools Palette. Custom objects can be added to the Tools Palette or created during run-time, with built-in style dialogs.
- Coordinate diagram entities with the Layout Manager, with just a mouse-click.
- Simplify user-navigation of large diagrams, with a zoomable Overview Window that synchronizes itself with the viewable portion of the canvas.

For detailed information, visit  
**www.infragistics.com**  
 Download Free Trial Version!  
 call 800-231-8588

- 1 Curved Relation Styles
- 2 Set colors and fonts for entities and relations
- 3 **NEW** List Entities
- 4 Overview Window with zoom control
- 5 Layout Manager and custom Tools Palette
- 6 **NEW** PropertyBrowser control included
- 7 User-defined Table Objects

**Infragistics™**  
 Component powered infrastructure

Copyright 2001 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo and InterAct are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

Java • ActiveX • .NET

800-231-8588 • **infragistics.com**



The Future Is Here With UltraGrid 2.0 Technology!

Flight_ID	WebSite	Shuttle	Departure_City	Arrival_City	Departure_Date	Arrival_Date
2023	http://www.infragistics.com	Apollo	New York, Earth	Aldrin, Jupiter	07/10/2027	07/10/2027

The launch of this flight is delayed by two hours. Please stand by for more information.

Flight_ID	Last_Name	First_Name	Row	Seat	Window	Class	Ticket
2023	Dak	Harold	D	5	<input checked="" type="checkbox"/>	Economy	Round Trip
2023	Pine	Charles	E	2	<input type="checkbox"/>	First	Round Trip

Flight_ID	WebSite	Shuttle	Departure_City	Arrival_City	Departure_Date	Arrival_Date
2119		Gemini	Chicago, Earth	Collins City, Mz	07/10/2027	07/10/2027

This flight is expected to launch on time.

Flight_ID	Last_Name	First_Name	Row	Seat
2119	Maple	Thomas	B	
2119	Adams	Joseph	A	

Flight_ID	WebSite	Shuttle	Departure_City
2231		Liberty	Los Angeles, B

No information exists for this flight.

5 Add... Flight\_ID Passengers

6

# More Control.

**Infragistics - formed by the merger of Sheridan and ProtoView**

Optimized for OLE DB technology, UltraGrid's unique features and components provide more control and flexibility than ever before. We've incorporated truly innovative graphical capabilities, giving the developer the ability to simplify navigation through complex data structures while providing eye-popping visual appeal. With new printing and print preview capabilities, displayed data can look great on paper, too.

## UltraGrid™ 2.0 Key Features:

- OLE DB bound
- Advanced non-bound modes
- Powerful per-Band data fetching options
- **NEW** Printing with Print Preview
- Pre-built user interfaces (ViewStyles) for flat and hierarchical representation of data
- Programmable objects provide accessibility at every level
- Cell, Row and Header transparency/translucency
- Built-in Masked Editing
- Background images for Grid, Rows, Cells, Headers, and more
- Built-in dropdown calendar
- Multiple Row and Column Scrolling Regions
- Intuitive context-sensitive hierarchical AddRow Bar
- Advanced Custom Property Pages with wizards
- Group, Column, Row and Cell objects
- Ability for owner-drawn UIElements
- **NEW** Save and Load Layouts Collection
- Value Lists
- **NEW** Data Formatting
- Automatic or Manual multi-column sorting of data



- 1 ViewStyle multiband vertical
- 2 Band Headers and Auto Preview Area
- 3 Display images and data in cells and headers
- 4 Dropdown calendar in grid cell
- 5 AddNewBox
- 6 Print Preview Screen

Seven powerful input components included with UltraGrid: ComboBoxEx, Currency, DateEdit, DropDownEdit, HotLink, Numeric and Spin.

Make the most of your investment—UltraGrid 2.0 is available with optional Subscription Service.\*

For detailed information, visit **www.infragistics.com**  
Download Free Trial Version!  
Order Online! or call 800-231-8588

**Infragistics™**  
Component powered infrastructure

Copyright 2001 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, and UltraGrid are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

Java • ActiveX • .NET

800-231-8588 • **infragistics.com**